



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo e implantación de una plataforma NFT de
bebidas de lujo

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Núñez Martí, Juan Vicente

Tutor/a: Gil Pechuán, Ignacio

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo e implementación de plataforma NFT de bebidas de lujo

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Juan Vicente Núñez Martí

Tutor: Ignacio Gil Pechuán

Curso 2021-2022

Resum

Si intentàrem mesclar una cosa tradicional i clàssic amb una cosa innovadora i d'actualitat aquest treball seria el resultat, és el cas d'ajuntar el món de les begudes de luxe i els NFTs.

En aquest projecte es desenvolupa una plataforma on es pot realitzar la compra d'aquestes begudes dins de la blockchain de Ethereum. El client el que compra és un NFT de la beguda en qüestió, amb el qual pot especular el seu preu, vendre quan desitge i, en el moment que considere, pot cremar-lo en la blockchain i l'empresa li enviarà aquesta beguda física.

Tot això es va desenvolupar mitjançant Solidity, llenguatge de programació orientat a contractes intel·ligents, i JavaScript.

Paraules clau: NFT, blockchain, luxe, plataforma

Resumen

Si intentásemos mezclar algo tradicional y clásico, con algo innovador y de actualidad este trabajo sería el resultado, es el caso de juntar el mundo de las bebidas de lujo y los NFTs.

En este proyecto se desarrolla una plataforma donde se puede realizar la compra de estas bebidas dentro de la blockchain de Ethereum. El cliente lo que compra es un NFT de la bebida en cuestión, con el que puede especular su precio, vender cuando desee y, en el momento que considere puede quemarlo en la blockchain y la empresa le enviará dicha bebida física.

Todo ello se desarrolló mediante Solidity, lenguaje de programación orientado a contratos inteligentes, y JavaScript.

Palabras clave: NFT, blockchain, lujo, plataforma

Abstract

If we tried to mix something traditional and classic with something innovative and current, this would be the result, it is the case of bringing together the world of luxury drinks and NFTs.

In this project, a platform is developed where you can make the purchase of these drinks within the Ethereum blockchain. What the client buys is an NFT of the drink in question, with which he can speculate on its price, sell when he wants and, at the moment he considers, he can burn it on the blockchain and the company will send him said physical drink.

All this was developed using Solidity, a programming language oriented to smart contracts, and JavaScript.

Key words: NFT, blockchain, luxury, platform

Índice general

Índice general	V
Índice de figuras	VII
Índice de tablas	VIII
<hr/>	
1 Introducción	1
1.1 Motivación	2
1.2 Objetivos	2
1.3 Impacto esperado	3
1.4 Metodología	4
1.5 Estructura de la memoria	4
1.6 Colaboraciones	4
2 Estado del arte	7
2.1 Estado del arte: valoración o versión crítica	9
2.2 Propuesta	10
3 Análisis del problema	11
3.1 Especificación de requisitos	11
3.1.1 Requisitos funcionales	13
3.1.2 Requisitos no funcionales	15
3.2 Solución propuesta	15
4 Diseño de la solución	17
4.1 Arquitectura del Sistema	17
4.2 Diseño Detallado	18
4.2.1 El contrato inteligente	18
4.2.2 El servidor	19
4.2.3 El marketplace	20
4.2.4 La plataforma	20
4.3 Tecnología Utilizada	20
4.3.1 Lenguajes de programación	20
4.3.2 Visual Studio Code	20
4.3.3 Remix	21
4.3.4 EtherScan	21
4.3.5 Webflow	21
4.3.6 OpenSea	21
5 Desarrollo de la solución propuesta	23
5.1 Creación y almacenamiento en la red de la metadata	23
5.2 Creación y lanzamiento del contrato inteligente	25
5.3 Listar la colección en el mercado secundario	29
6 Implantación	33
7 Pruebas	37
7.1 Iniciar sesión	38
7.2 Comprar NFT a través de una billetera	40
7.3 Comprar NFT a través de la cuenta del banco	42

7.4 Visualizar los NFTs	43
8 Conclusiones	45
8.1 Relación del trabajo desarrollado con los estudios cursados	45
8.2 Trabajos futuros	46
Bibliografía	47

Apéndice

A Objetivos de desarrollo sostenible	49
---	-----------

Índice de figuras

1.1	Blockchain	1
1.2	NFT de la colección Bored Ape Yacht Club	2
1.3	Comparación entre bebidas de lujo y el S&P 500	3
2.1	Primer NFT de la historia	7
2.2	CryptoPunk más caro vendido	8
2.3	BlockBar	9
3.1	Diagrama de casos de uso	12
4.1	Diagrama de la arquitectura del sistema	17
4.2	Diagrama visualización NFT	18
4.3	Archivos dentro del servidor	19
5.1	Imagen de los NFTs	23
5.2	Contenido dentro de los JSON	24
5.3	Proyecto que se sube al servidor	24
5.4	Consola	25
5.5	Variables del contrato inteligente	26
5.6	Función de mintear	26
5.7	Funciones privadas	27
5.8	Deploy	27
5.9	Transacción aceptada	28
5.10	Detalles de la transacción	28
5.11	Detalles del contrato	29
5.12	Listar colección en OpenSea	30
5.13	Colección listada en OpenSea	30
6.1	Constantes de la plataforma	33
6.2	Importar librería Web3	34
6.3	Actualización de variables	34
6.4	Función que comprueba la red en la que te encuentras	34
6.5	Función de mintear en la plataforma	35
6.6	Código relacionado con Crossmint	35
7.1	Diseño de la plataforma	37
7.2	Barra de navegación - Login	38
7.3	Inicio de sesión	38
7.4	MetaMask	39
7.5	Advertencia cambiar de red	39
7.6	Botón para mintear	40
7.7	Transacción MetaMask	40
7.8	Transacciones del contrato	41
7.9	Perfil en OpenSea	41
7.10	Botón para comprar a través del banco	42

7.11 Pasarela de pago	42
7.12 Cartera en Crossmint	43
7.13 Ver NFTs de una billetera	44

Índice de tablas

3.1 Requisito funcional 1, registrarse con correo electrónico.	13
3.2 Requisito funcional 2, registrarse con billetera.	13
3.3 Requisito funcional 3, iniciar sesión.	14
3.4 Requisito funcional 4, comprar NFT.	14
3.5 Requisito funcional 5, visualizar inventario.	14
3.6 Requisito funcional 6, quemar NFT.	14
3.7 Requisito no funcional 1, portabilidad.	15
3.8 Requisito no funcional 2, rendimiento.	15
3.9 Requisito no funcional 3, mantenibilidad.	15

CAPÍTULO 1

Introducción

En la actualidad si hay un tema que está en boca de todos, ese es sin duda: Blockchain. Y evidentemente todo lo relacionado con ello, empezando por las criptomonedas, pasando por el mundo de las finanzas descentralizadas, y acabando por la trazabilidad que puede ofrecer esta tecnología.

Podemos entender el término blockchain como “una gran cadena de datos cuya información se agrupa en bloques, donde cada bloque contiene información relativa a otro, por lo que solo podríamos editar la información de un bloque modificando todos los bloques anteriores.”¹

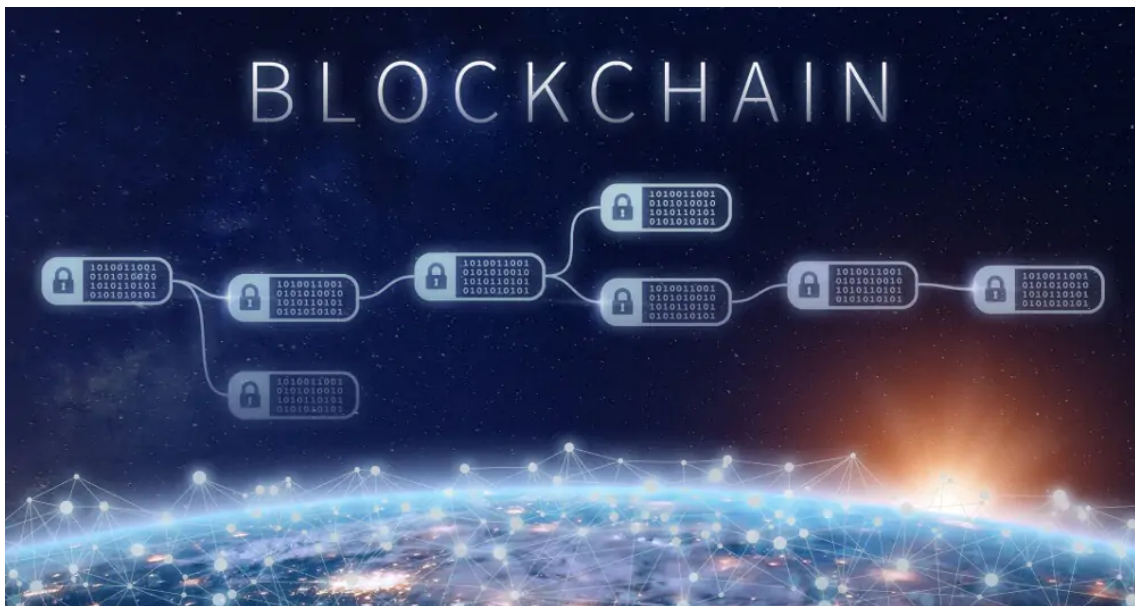


Figura 1.1: Blockchain. (Imagen extraída de ²)

Por ello, una de las grandes características que ofrece, es la seguridad, ya que la información se encuentra distribuida, encriptada y compartida entre los nodos de la red, y que para operar sobre ella se necesita tener un conjunto de claves, por lo que resulta muy difícil de hackear.

Esta tecnología ha despertado la mente de muchas personas para poder beneficiarse de este gran auge que está teniendo. Comenzó con el tema de las criptomonedas, que tiene un gran parecido a la bolsa, y ahora este auge está trasladándose a los NFTs.

¹https://es.wikipedia.org/wiki/Cadena_de_bloques

²<https://www.iebschool.com/blog/blockchain-cadena-bloques-revoluciona-sector-financiero-finanzas/>

NFT es la abreviación de Non Fungible Token, es decir, un token que no puedes tocar. Es muy similar a comprar un token de una criptomoneda, pero lo que estás comprando es arte. Puede ser una foto, un meme, una canción...

Cuando compras un NFT, lo que estás comprando es la propiedad de esa pieza de arte, todo el mundo podría descargarse tu NFT pero solo tú serías el dueño, y esto se consigue gracias a que se utiliza la tecnología blockchain.

Actualmente existen muchas colecciones de NFTs valoradas en millones de euros, como puede ser el caso de: Bored Ape Yacht Club, cuyo token más barato se encuentra a día de hoy valorado en 145 ETH, es decir, 404.550€.



Figura 1.2: NFT más barato de la colección Bored Ape Yacht Club. (Imagen extraída de ¹)

En este trabajo se plantea realizar un proyecto basado en esta tecnología donde creo una colección de NFTs basada en el mundo de las bebidas de lujo y la implementaré en una plataforma que programaré para su venta.

1.1 Motivación

Desde hace un año que estoy dentro del mundo de la blockchain, siempre desde un punto de vista profesional (como emprendedor). Gracias a la oportunidad que me ofreció mi tutor y mi compañero de TFG puedo sumergirme hasta la raíz del código para poder entender mucho mejor su funcionamiento.

Además de ser un TFG, deseo que sea un proyecto, y posteriormente de su realización y redacción, espero poder lanzar dicha colección para finales de este año, y que todo ese esfuerzo que dedique en este trabajo dé sus frutos.

1.2 Objetivos

El objetivo principal es desarrollar una plataforma que permita a los usuarios poder realizar la compra de nuestra colección limitada de NFTs de bebidas de lujo, a través de

¹<https://opensea.io/assets/0xbc4ca0eda7647a8ab7c2061c2e118a18a936f13d/9719?ref=0xb233ddab5da16808a2401b6895e129f4854e2744/>

sus billeteras en la red de Ethereum. Dicha plataforma está enfocada tanto para aquellos compradores experimentados, como para los principiantes, y únicamente precisarían de una billetera caliente.

Los objetivos específicos que debería cumplir la plataforma son los siguientes:

- Llevar a cabo una interfaz que sea fácil e intuitiva para el usuario principiante.
- El usuario podrá registrarse a través de su billetera o correo electrónico.
- Si a la hora de realizar la compra, la billetera no se encuentra en la red de Ethereum la plataforma se lo notificará.
- Permitir realizar la compra a través de una transacción bancaria.
- Una vez el usuario haya adquirido su NFT siempre tendrá la opción de quemarlo.
- Existirá una página donde permitirá visualizar al usuario los NFTs que le pertenecen.

1.3 Impacto esperado

No quiero hacer una colección como todas las demás, quiero ir un poco más allá del resto. En la actualidad vemos colecciones vendidas por artistas por una gran cantidad de dinero, y al fin y al cabo, se tratan de un simple JPG, cuyo único beneficio es poder revenderlo más caro.

Por comprar y mantener nuestros NFTs, los usuarios tendrán diversas ventajas. Dividiremos estas ventajas en dos partes, una primera parte física, relacionada con la bebida de lujo, y una segunda parte relacionada con el sector tecnológico.

Vayamos con las primeras ventajas relacionadas con las bebidas, sabemos que podemos invertir en muchos sectores, uno de ellos es el mundo de las bebidas de lujo, donde podemos ver en la siguiente gráfica como ha evolucionado su precio en comparación con otros activos.

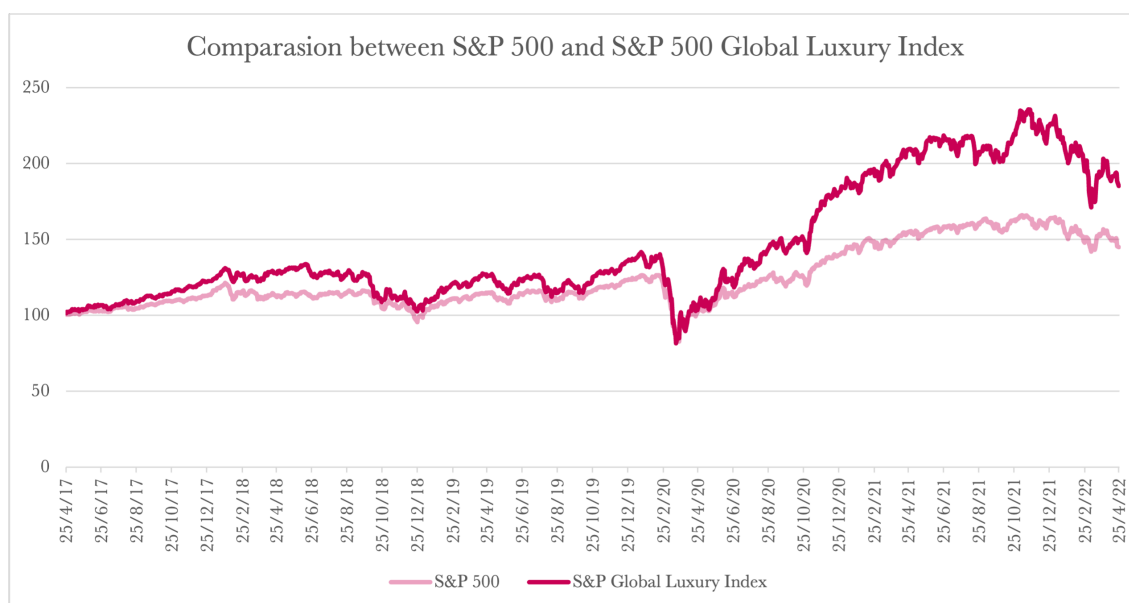


Figura 1.3: Comparación entre bebidas de lujo y el S&P 500 (Autor: Pedro Olivares, 2022).

He aquí su primera ventaja, nosotros te ofrecemos un servicio de inversión, donde además te conservaremos la bebida en el mejor estado posible, y eliminaremos todas las posibilidades en la que desafortunadamente podrían deteriorar dicha bebida en tu hogar.

Desde el punto de vista tecnológico tendrás acceso a una comunidad que se formará entre todo los compradores, donde de manera periódica se regalarán productos de la marca, se realizarán eventos presenciales (como catas de vino), se otorgarán privilegios para una futura colección...

1.4 Metodología

Lo primero de todo va a ser realizar un proceso de recopilación de colecciones de NFTs para poder investigarlas y poder visualizar sus puntos fuertes y débiles, y determinar cuales de sus características han permitido que dicha colección sea un éxito.

Revisando cada parte del proyecto, desde su elaboración desde el código fuente, hasta su marketing y venta, para poder ver que tecnologías han utilizado para su elaboración y difusión.

Una vez realizado esto, estudiar esa parte tecnológica para poder desarrollar nuestro proyecto, teniendo ya claro los elementos que vamos a utilizar para ello.

1.5 Estructura de la memoria

El trabajo consta de los siguientes capítulos:

- **Capítulo 1:** Introducción del tema que vamos a abordar, establecer los objetivos que se intentan cumplir con el proyecto, y colaboraciones realizadas.
- **Capítulo 2:** Esclarezco el contexto de la situación tecnológica en la que nos encontramos, realizo una crítica y establezco las mejoras que voy a realizar.
- **Capítulo 3:** Planteo el problema, indico como lo voy a tratar y ordeno todos los requisitos que ha de tener la solución final.
- **Capítulo 4:** Comienzo con el planteamiento del diseño de la solución, donde explico que elementos intervienen y como interactúan entre ellos.
- **Capítulo 5:** Desarrollo la solución anteriormente propuesta.
- **Capítulo 6:** Implemento en nuestra plataforma la solución que he desarrollado.
- **Capítulo 7:** Realizo una serie de pruebas para comprobar el correcto funcionamiento de las funcionalidades más relevantes de la plataforma.
- **Capítulo 8:** Sintetizo el trabajo desarrollado en una conclusión final y menciono cuales serán mis trabajos futuros.

1.6 Colaboraciones

Este Trabajo Fin de Grado se realizó junto a mi compañero Pedro Olivares Sanchez, donde él se dedicó más al apartado de la empresa: análisis de requisitos, viabilidad, análisis de mercado... También es el encargado de dar el diseño a la plataforma.

En cambio mi trabajo es la creación de dicha colección en la red de Ethereum a través de contratos inteligentes, e implementarlo en la plataforma.

CAPÍTULO 2

Estado del arte

Los NFTs no es algo de este año o de hace dos, “hace una década ya se hablaba de un concepto denominado: moneda de colores. Estas monedas representarían activos no tangibles, como puede ser el caso de Bitcoin, en activos que si lo son.”¹

Es cierto que este concepto queda muy lejos de lo que a día de hoy conocemos como NFT, pero si que sirvió de inspiración para lo que vendría después.

“El 2 de Mayo de 2014 fue creado Quantum por el artista digital neoyorquino Kevin McCoy. Esta pieza es considerada como el primer NFT, el primer ejemplo del criptoarte.”² Dicha obra se subastó y finalmente se vendió por casi 1,5 millones de dólares.

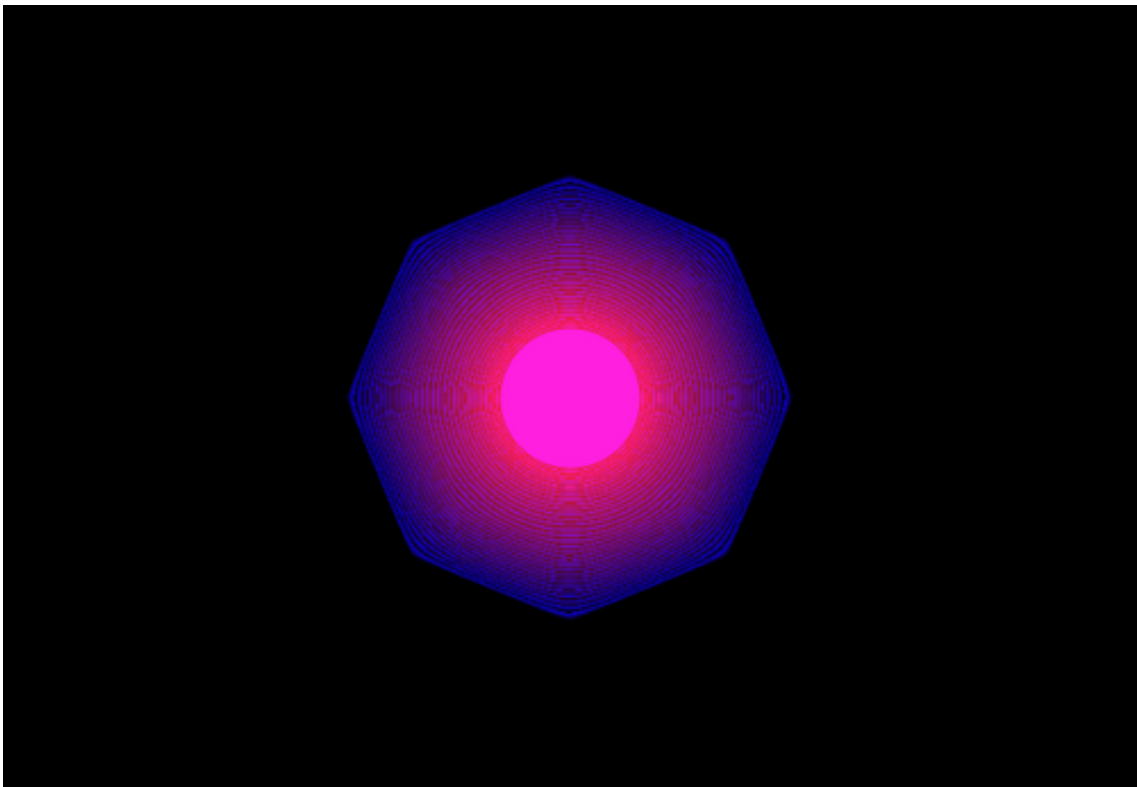


Figura 2.1: Quantum, el primer NFT de la historia (Imagen extraída de ³).

¹<https://www.knowyourmobile.com/es/noticias/cuando-empezó-nfts/>

²<https://elemental.com/2021/06/10/el-primer-nft-de-la-historia-vendido-por-casi-15-millon-de-dolares-en->

³<https://www.coindesk.com/news/articleView.html?idxno=77217>

No es hasta 2015 (cuando la cadena de bloques de Ethereum se puso en marcha) cuando el mundo de los NFTs iba a revolucionar. Esta cadena permitía a los demás usuarios crear contratos inteligentes, y por consiguiente sus NFTs.

Si pensamos en una primera colección donde se empezaron a mover las primeras grandes cantidades de dinero, sin duda estamos pensando en CryptoPunks. Colección de 10.000 NFTs lanzada en junio de 2017, donde hasta la fecha el CryptoPunk más caro se vendió por 8.000 ETH, es decir, más de 21 millones de dólares.



Figura 2.2: CryptoPunk más caro vendido (Imagen extraída de ¹).

Siete años después de esta revolución han pasado muchas colecciones por distintas cadenas de bloques, y como era evidente esto no se ha quedado aquí, sino que ha evolucionado. La mayoría de estas, ya no solo venden el propio NFT sino que además proporcionan una serie de beneficios para los dueños, o tienen una funcionalidad extra por poseer dicha pieza artística.

En este proyecto nos vamos a centrar en la venta de bebidas de lujo a través de colecciones de NFTs. Es un mercado todavía no explotado, después de una gran búsqueda tan solo hemos podido encontrar una única plataforma donde se realice todo esto de forma profesional y fructífera.

Esta plataforma se denomina BlockBar, y realiza diversas colaboraciones con grandes marcas de bebidas de lujo en forma de colecciones de NFTs. Lo que el cliente compra es un NFT, pero está respaldado por una botella. El cliente puede especular con el precio, y vender cuando desee, además cuando él decida puede quemar dicho NFT, y la plataforma le enviará dicha botella.

¹<https://cryptopunks.app/cryptopunks/details/5822>

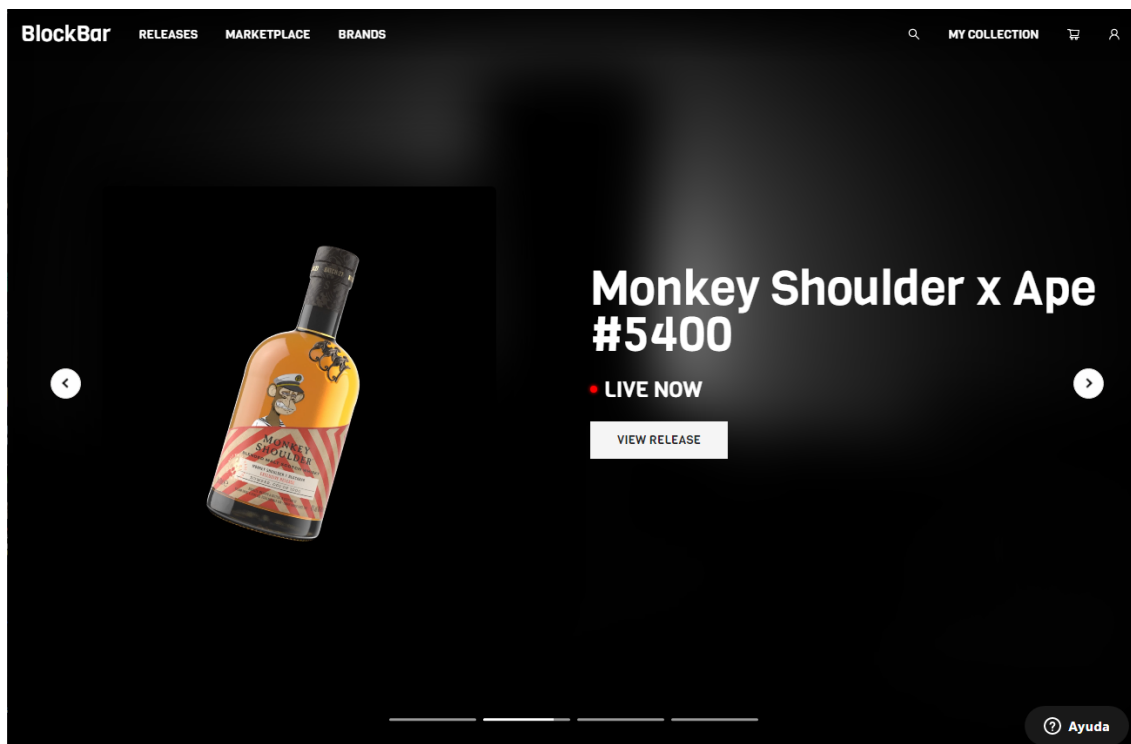


Figura 2.3: BlockBar (Visitada en Junio, 2022).

Cabe señalar que las colecciones de estas bebidas exclusivas solo se podrán comprar a través de la plataforma, por la que la única manera de adquirir esa botella física es a través de BlockBar.

Para el cliente realizar la transacción de esta forma tiene los siguientes beneficios:

- Puede especular con el precio, en el caso de que se devaluara dicho NFT, siempre tendrá el respaldo de la botella en cuestión.
- Desde la plataforma custodian la botella, cosa que si tu la compraras y la almacenaras en tu hogar, siempre tendrías el riesgo de poder perder dicha adquisición.
- Se trata de una forma más de poder invertir el capital, en un sector que vemos cada vez más en auge.

Y es evidente los beneficios que aportan dichas colecciones tanto a la empresa colaborada como a la plataforma. Al tratarse de un elemento de lujo y exclusivo, se le puede establecer el valor que uno desee.

2.1 Estado del arte: valoración o versión crítica

En la actualidad podemos observar como salen a la luz una gran cantidad de colecciones, algunas de ellas con proyectos serios, pero la mayoría de estas no valen la pena debido a que sus creadores buscan dinero rápido y fácil.

Con diseños realizados en Paint, debido a que todo su capital de inversión lo destinan a captar personas con influencia para persuadir a usuarios a comprar dicha colección. Y

¹<https://cryptopunks.app/cryptopunks/details/5822>

esto es normal que suceda, como sucede en todos los sectores que están en pleno crecimiento.

Nosotros queremos formar parte del grupo de proyectos serios y con buenos fundamentales. Como podría ser el caso de BlockBar, del cual potenciaremos sus puntos fuertes y reduciremos sus aspectos negativos con nuestra propuesta.

2.2 Propuesta

En este trabajo final de carrera se pretende lanzar una plataforma donde sea a la vez una carta de presentación a nuestros clientes y además donde se pueda realizar la primera compra de una colección de bebidas de lujo exclusiva.

Nos basaremos en BlockBar pero realizaremos un cambio muy significativo y distintivo a la hora de generar ingresos, es decir, replantearemos a nivel empresarial la manera de obtención de beneficios.

El mayor beneficio que se genera, es cuando se vende por primera vez un NFT de una colección, pero también se puede generar beneficio estableciendo unos royalties en cada transacción, nosotros estableceremos estos de la siguiente forma: Un cinco por cien para nuestra plataforma, y otro cinco para la empresa colaboradora, cuando esta empresa visualice que cuanto más volumen, es decir, más transacciones se produzcan, más dinero generarán, se interesarán más en que haya más movimiento en su colección.

Lo que se intenta lograr con este cambio, es que la empresa colaboradora no se olvide del cliente una vez haya comprado su producto, sino que lo cuide y le otorgue ciertas ventajas, para que otros usuarios que aún no sean clientes deseen serlo, y por consecuente haya más transacciones.

Estos royalties no solo benefician a la empresa colaboradora, sino también a nuestra empresa como plataforma, ya que tenemos también un porcentaje de cada transacción. Este cambio que presentamos creemos que aporta más valor a todos los clientes y entidades participantes.

Aunque esta idea de negocio está pensada para realizar colaboraciones con marcas de éxito, para este trabajo sacaremos una colección de vinos exclusiva de la bodega de mi compañero Pedro.

CAPÍTULO 3

Análisis del problema

En este capítulo vamos a comentar esas oportunidades que hemos visto, y que existen, gracias al estudio que acabamos de realizar al estado del arte.

Para ello vamos a realizar una descripción completa de la plataforma a través de la especificación de requisitos.

3.1 Especificación de requisitos

La plataforma una vez terminada deberá cumplir con las siguientes funcionalidades:

- Debe posibilitar la conexión con la billetera del usuario, reconocer en que red se encuentra, y si no está en la adecuada, solicitar que se cambie a la red en la que la colección se encuentre.
- Debe permitir al usuario poder realizar la compra de NFTs de la colección, siempre y cuando controle el número de tokens que falten por vender y que la cantidad de la transacción no supere el número máximo permitido de compra. Con esto se intenta evitar que la colección pueda ser manipulada por una persona, cuanto más dividido esté, menos probabilidad existe de manipulación de mercado.
- Debe posibilitar la opción de compra a través de tarjeta bancaria, ya que no queremos cerrar las puertas a aquellos posibles compradores que no estén muy familiarizados todavía con este mundo.
- Ha de contar con un sistema de registro, ya sea a través de un correo electrónico o billetera caliente del usuario.
- Todos aquellos usuarios que se hayan registrado e iniciado sesión podrán visualizar sus NFTs, desde donde podrán gestionar las acciones que deseen realizar con el mismo.
- Debe permitir la quema de los tokens a aquellos usuarios que se hayan registrado y sean dueños de algún NFT, esto significa retirar el token de la circulación y enviar la bebida respaldada al dueño.

Una vez realizado la especificación de requisitos vamos a incluir todas las funciones, para ello he realizado el siguiente diagrama de casos de uso. En este diagrama se reflejan todas las acciones que puede realizar un usuario en nuestra plataforma.

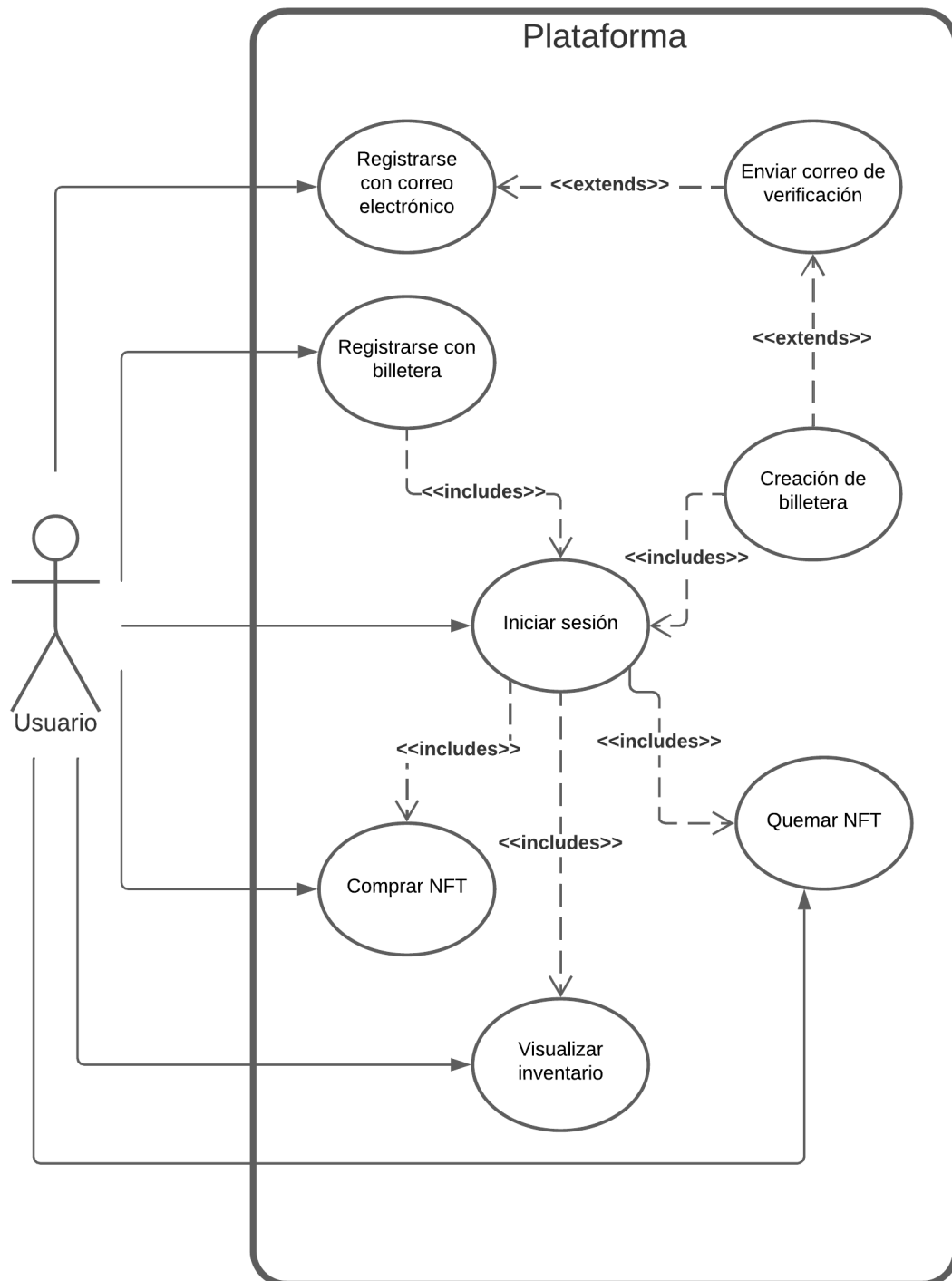


Figura 3.1: Diagrama de casos de uso (Elaboración propia).

- **Registrarse con correo electrónico:** El usuario tiene la posibilidad de registrarse a través de su correo electrónico. El sistema le enviará un correo de verificación y cuando el usuario lo acepte, se generará de manera automática una billetera.
- **Registrarse con billetera:** Si se trata de un usuario más experimentado, podrá iniciar sesión con la billetera MetaMask para poder realizar el registro.
- **Iniciar sesión:** Una vez el usuario ya se haya registrado, ya sea a través del correo o de una billetera ya existente, podrá iniciar sesión en la plataforma.
- **Comprar NFT:** El usuario podrá comprar NFTs de la colección, siempre y cuando no supere el número máximo permitido por billetera.
- **Visualizar inventario:** La plataforma proporciona una ventana para que el usuario pueda visualizar los NFTs que le pertenecen.
- **Quemar NFT:** Si un usuario se ha registrado en la plataforma y es dueño de un NFT de la colección, la plataforma debe posibilitar al dueño la opción de la quema del token.

3.1.1. Requisitos funcionales

Los requisitos funcionales que definen las funciones de la plataforma son las siguientes:

Identificador del requisito	RF-01
Nombre del requisito	Registrarse con correo electrónico
Requisito relacionado	-
Descripción del requisito	El usuario podrá registrarse en nuestra plataforma a través del correo electrónico
Prioridad del requisito	Alta
Precondición	-
Postcondición	-

Tabla 3.1: Requisito funcional 1, registrarse con correo electrónico.

Identificador del requisito	RF-02
Nombre del requisito	Registrarse con billetera
Requisito relacionado	-
Descripción del requisito	El usuario podrá registrarse en nuestra plataforma a través de su billetera de criptomonedas
Prioridad del requisito	Alta
Precondición	-
Postcondición	-

Tabla 3.2: Requisito funcional 2, registrarse con billetera.

Identificador del requisito	RF-03
Nombre del requisito	Iniciar sesión
Requisito relacionado	RF-01, RF-02
Descripción del requisito	El usuario podrá iniciar sesión en nuestra plataforma
Prioridad del requisito	Alta
Precondición	El usuario tendrá que haberse registrado previamente
Postcondición	-

Tabla 3.3: Requisito funcional 3, iniciar sesión.

Identificador del requisito	RF-04
Nombre del requisito	Comprar NFT
Requisito relacionado	RF-03
Descripción del requisito	El usuario podrá comprar NFTs de la colección
Prioridad del requisito	Esencial
Precondición	El usuario tendrá que haber iniciado sesión previamente
Postcondición	-

Tabla 3.4: Requisito funcional 4, comprar NFT.

Identificador del requisito	RF-05
Nombre del requisito	Visualizar inventario
Requisito relacionado	RF-03
Descripción del requisito	El usuario podrá ver los NFTs de la colección que haya comprado
Prioridad del requisito	Media
Precondición	El usuario tendrá que haber iniciado sesión previamente
Postcondición	-

Tabla 3.5: Requisito funcional 5, visualizar inventario.

Identificador del requisito	RF-06
Nombre del requisito	Quemar NFT
Requisito relacionado	RF-03
Descripción del requisito	El usuario podrá quemar los NFTs de la colección que haya comprado
Prioridad del requisito	Alta
Precondición	El usuario tendrá que haber iniciado sesión previamente
Postcondición	-

Tabla 3.6: Requisito funcional 6, quemar NFT.

3.1.2. Requisitos no funcionales

A continuación explicaremos aquellos requisitos que explican aspectos de calidad del sistema:

Identificador del requisito	RNF-01
Nombre del requisito	Portabilidad
Descripción del requisito	La plataforma ha de funcionar correctamente independientemente del navegador que se esté utilizando
Prioridad del requisito	Alta

Tabla 3.7: Requisito no funcional 1, portabilidad.

Identificador del requisito	RNF-02
Nombre del requisito	Rendimiento
Descripción del requisito	La plataforma ha de cargarse en menos de dos segundos
Prioridad del requisito	Media

Tabla 3.8: Requisito no funcional 2, rendimiento.

Identificador del requisito	RNF-03
Nombre del requisito	Mantenibilidad
Descripción del requisito	El código de la plataforma ha de tener únicamente las líneas necesarias y todo debidamente comentado con su funcionalidad
Prioridad del requisito	Media

Tabla 3.9: Requisito no funcional 3, mantenibilidad.

3.2 Solución propuesta

La solución será desarrollar una plataforma web en la que se pueda realizar la compra de NFTs de nuestra colección de bebidas de lujo.

Para ello deberemos seguir los siguientes pasos: Primeramente deberemos crear dicha colección, es decir, las imágenes y su metadata. Una vez creado, lo subiremos al servidor que hayamos escogido para almacenar dichos NFTs.

Después trabajaremos con el contrato inteligente. Aquí es donde crearemos todas las funcionalidades que tendrá la colección, y actuará como una cuenta bancaria, donde se recibirá todo lo recaudado.

Una vez dicho contrato lo lancemos a la red de Ethereum, ya podremos listar nuestra colección en los marketplaces más relevantes en el mundo de los NFTs, como es el caso de OpenSea.

Por último crearemos nuestra plataforma donde se pueda realizar el minteo de nuestra colección, el minteo es la primera compra de un NFT de la colección. Donde esta tendrá todas las funcionalidades anteriormente mencionadas.

CAPÍTULO 4

Diseño de la solución

El proyecto desarrollado consta de cuatro pilares: El contrato inteligente, el servidor donde almacenaremos la metadata de cada NFT, el marketplace donde se realizará la venta secundaria y la plataforma donde realizaremos la primera venta pública (también denominado minting).

4.1 Arquitectura del Sistema

Estos cuatro pilares funcionan de la siguiente forma: el marketplace está conectado al contrato inteligente, y este a su vez lo está con el servidor que contine la metadata de la colección, cuando dentro del marketplace se tenga que efectuar alguna visualización o transferencia de nuestra colección, mandará una solicitud al contrato inteligente y este la tratará, si fuera necesario el contrato recuperará la metadata del servidor y este la reenviará al marketplace.

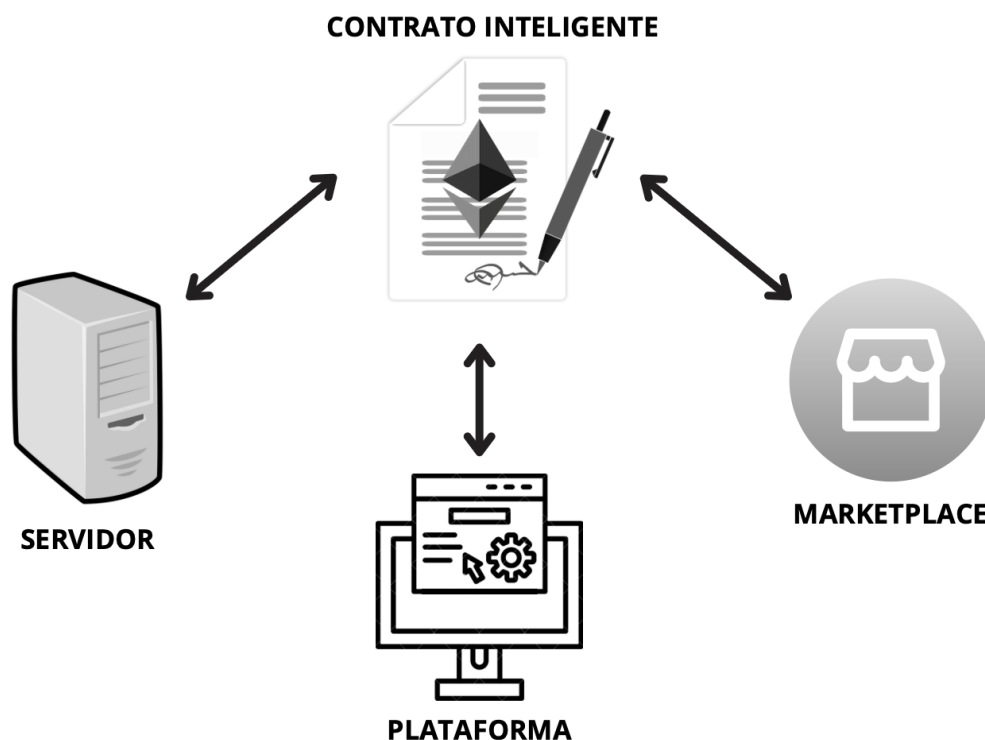


Figura 4.1: Diagrama de la arquitectura del sistema (Elaboración propia).

Como ya hemos comentado la plataforma será nuestro escaparate para los futuros clientes, pero es allí también donde se efectuarán las primeras compras de nuestra colección, por lo que la plataforma estará conectada con las funciones del contrato inteligente relacionadas con esa primera compra.

Un ejemplo posible sería el siguiente: Un usuario del marketplace quiere visualizar un NFT. El marketplace interactuará con el contrato inteligente, y este le solicitará la metadata al servidor, el servidor se la enviará al contrato inteligente y este al marketplace.

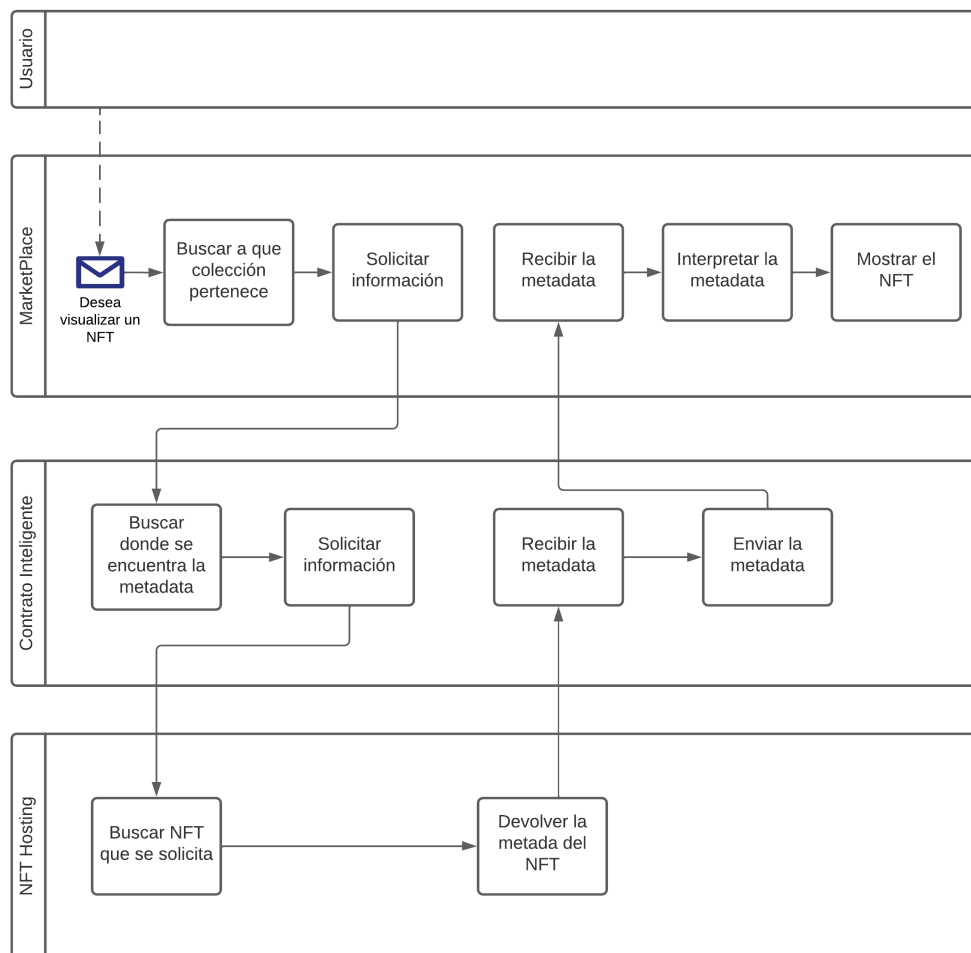


Figura 4.2: Diagrama visualización NFT (Elaboración propia).

4.2 Diseño Detallado

Una vez conocido todos los elementos que van a participar en este trabajo, procedemos a detallarlos uno a uno.

4.2.1. El contrato inteligente

Lo primero de todo es crear, implementar y lanzar una clase en la blockchain de Ethereum. Básicamente se trata de un programa que se ejecuta en dicha blockchain, donde cuyas funciones y estados existen en una dirección específica que sea crea al lanzar la clase.

Tan solo necesitaremos de una clase, con la que contaremos con un número reducido de funciones importantes, más adelante explicaremos aquellas que sean más relevantes para el correcto funcionamiento del proyecto.

La clase que hemos lanzado se convertirá en un contrato inteligente, que actuará como si fuera una cuenta bancaria, es decir, se pueden realizar y recibir transacciones. Pero esta cuenta no está controlada por ningún usuario, y efectúan todos sus movimientos acorde lo programado previamente.

Este contrato inteligente actuará como intermediario entre el marketplace donde se venderá nuestra colección y el servidor donde tendremos almacenado toda la información relativa a nuestros NFTs. También tendrá otra función relacionada con la plataforma, que explicaremos más adelante.

4.2.2. El servidor

En el servidor subiremos tres tipos de archivos:

- **Imágenes de los NFTs:** En nuestro caso cada NFT tendrá la misma imagen, ya que las botellas tendrán el mismo diseño. Y en vez de subir la misma imagen tantas veces como NFTs tengamos, podemos utilizar la misma para cada NFT.
- **Archivos JSON para almacenar la información de cada NFT:** Aquí se almacenará toda la metadata de cada NFT, es decir, su número identifiativo, su descripción, la imagen que tiene, sus atributos... Profundizaremos más adelante como hemos generado de forma automática todos los JSON.
- **ABI:** Son las siglas de Application Binary Interface, es el resultado de codificar nuestro contrato inteligente para que nuestra plataforma pueda interactuar con el contrato. Se trata de un JavaScript donde se almacenan todas las funciones que contiene nuestro contrato.

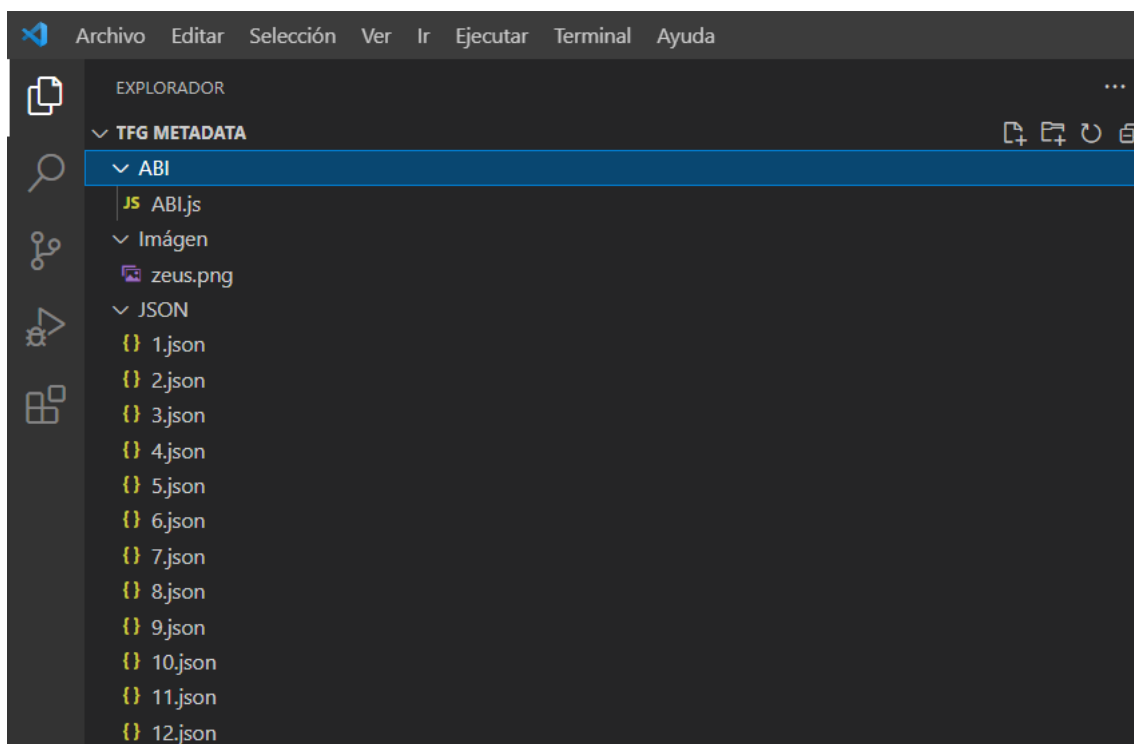


Figura 4.3: Archivos dentro del servidor (Elaboración propia).

4.2.3. El marketplace

También necesitaremos de un mercado para la comercialización de NFTs. Un lugar donde todo lo creado anteriormente tenga sentido, es decir, que la colección pueda venderse y comercializarse.

El marketplace estará conectado a nuestro contrato inteligente, ya que necesitará todos los atributos relacionados con cada NFT para poder mostrárselo a posibles compradores.

Podemos entenderlo como la parte visible, como una interfaz. La cuál es mucho más intuitiva y amigable para el usuario.

4.2.4. La plataforma

La plataforma será nuestro escaparate profesional de cara al público, donde explicaremos cuál es el equipo que hay detrás del proyecto, cuál va a ser la colección que lancemos, insertaremos vídeos para generar bombo publicitario...

Pero realmente la funcionalidad más importante de la plataforma será el minteo, es decir, la primera compra de cada uno de los NFTs se realizará desde esta. Por lo que deberá estar conectado con nuestro contrato inteligente.

4.3 Tecnología Utilizada

4.3.1. Lenguajes de programación

- **Solidity:** Solidity es un lenguaje de programación de alto nivel orientado a objetos para crear contratos inteligentes. Es similar a otros lenguajes como C y JavaScript. Fue desarrollado por colaboradores de Ethereum en 2014.

Como principal ventaja encontramos que el código fuente de los contratos inteligentes pueden ser accedidos públicamente desde la blockchain.

Como desventaja, pueden ser muy vulnerables si contienen errores en el código, ya que si no lo programáramos de manera correcta podríamos llegar a provocar el robo de la cantidad que hubiera dentro.

- **JavaScript:** Se trata de un lenguaje de programación ligero, interpretado y se define como un lenguaje orientado a objetos.

Las grandes ventajas con las que cuenta son: su rapidez, simplicidad y compatibilidad, ya que podemos utilizarlo en diferente tipos de aplicaciones.

En contraposición, podemos encontrar la seguridad, ya que los fallos podemos explotarlos para malos propósitos. También, dependiendo del navegador que utilicemos el script puede mostrar no siempre el mismo resultado.

4.3.2. Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft totalmente gratuito y de código abierto.

Se utilizó para importar en un proyecto todo lo necesario, para después utilizar la consola y hacer un 'deploy' al servidor donde se almacena toda la información que necesitaremos más adelante en otras plataformas.

4.3.3. Remix

Remix es un IDE (Entorno Integrado de Desarrollo) que se puede ejecutar a través de un navegador ya que integra un compilador y un entorno en tiempo de ejecución para Solidity.

Se utilizó dicho entorno para crear, desarrollar y lanzar el contrato inteligente en la red de Ethereum.

4.3.4. EtherScan

EtherScan es una plataforma web para realizar un seguimiento de las transacciones, visualizar contratos inteligentes y, en general, mantenerse al día con lo que sucede en Ethereum.

Desde esta plataforma podemos visualizar si nuestro contrato se ha creado con éxito, cuantas ventas se han realizado de nuestra colección, que billeteras tienen nuestros NFTs, el precio por el que se están pagando...

Es una herramienta muy valiosa para nosotros, ya que podemos hacer un seguimiento muy detallado de nuestro proyecto.

4.3.5. Webflow

Webflow es una herramienta que permite el diseño y gestión de contenido web, al tener un constructor visual permite crear páginas web con una filosofía no-code.

Elegimos esta herramienta para nuestra plataforma porque se pueden realizar diseños muy profesionales, sin tener un gran conocimiento en CSS.

A parte también, se puede escribir código, y para nosotros esta parte era imprescindible, ya que nuestra plataforma ha de estar conectada con nuestro contrato inteligente, ya que se podrá hacer compras de nuestra colección en la plataforma.

4.3.6. OpenSea

OpenSea es el mayor mercado de NFTs en la actualidad, fue desarrollada en 2017. En dicha plataforma se pueden realizar compras y ventas en la red Ethereum, Solana y Matic.

Decidimos que esta plataforma debería ser nuestro mercado secundario, es decir, después de la primera compra todas las transacciones se realizarían en OpenSea. Ya que es donde se produce el mayor volumen de compra/venta de NFTs en la red de Ethereum.

Para listar nuestra colección basta con proporcionar nuestra dirección dentro de Ethereum, y automáticamente la plataforma ya detecta toda la colección asociada a ella a través de nuestro contrato inteligente.

CAPÍTULO 5

Desarrollo de la solución propuesta

En este capítulo vamos a describir el desarrollo de la solución propuesta, comentando que problemas y dificultades hemos tenido en su realización, y las decisiones que hemos tenido que ir tomando según trabajábamos en la plataforma.

Por motivos de seguridad y propiedad intelectual, el número de NFTs, el nombre y la metadata de la colección que se desarrolla en este punto, no coinciden con el proyecto final. Cabe destacar que el proceso y el desarrollo es el mismo.

Para este trabajo creamos una colección de cincuenta tokens, bajo el nombre de "Zeus", donde cada NFT tiene un valor de 0.02 ETH. Y lo lanzamos en la red de pruebas de Ethereum, Rinkeby, con el objetivo de ahorrarnos los grandes costes de su red principal.

Todos los NFTs de nuestra colección tienen esta imagen:



Figura 5.1: Imagen de los NFTs (Elaboración propia).

5.1 Creación y almacenamiento en la red de la metadata

Como hemos explicado anteriormente, la metadata es toda la información que posee un NFT. En esta sección asignamos cada NFT con la suya en archivos JSON, en este caso van a compartir todos los tokens los mismos atributos (imagen, descripción...) excepto el nombre, el cuál será distinto en cada uno.

Para ello creamos un proyecto dentro de Visual Studio Code y generamos tantos archivos JSON como NFTs tiene la colección, es decir, cincuenta.

Dentro de los JSON es donde introducimos toda la información relacionada con el NFT. La información que nosotros hemos escogido que almacene es la siguiente:

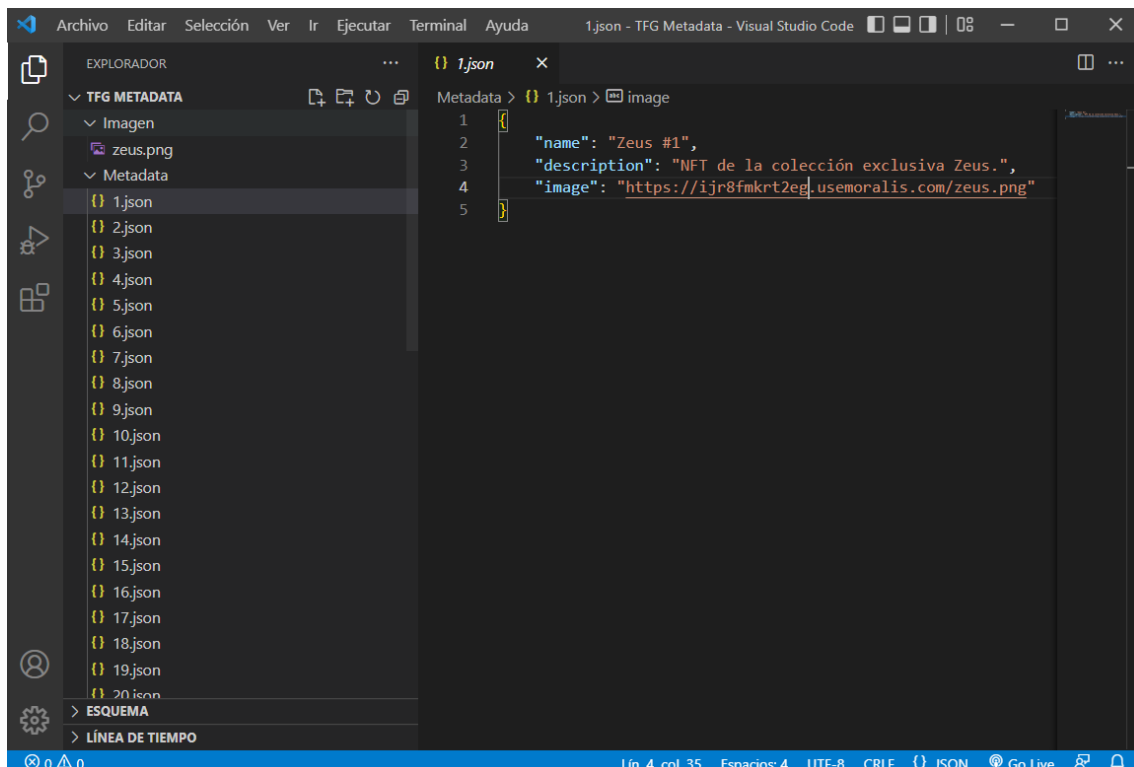
- **Nombre:** Tiene el nombre de la colección acompañado por un hastag y por el número que tiene dentro de la colección.
- **Descripción:** Se trata de una breve descripción del token.
- **Imagen:** Insertamos el enlace donde se encuentre la imagen dentro del servidor, más adelante explicaremos como realizamos esta subida.

```
{ } 1.json x
{ } 1.json > image
1  {
2    "name": "Zeus #1",
3    "description": "NFT de la colección exclusiva Zeus.",
4    "image": "https://ijr8fmkrt2eg.usemoralis.com/zeus.png"
5  }
```

Figura 5.2: Contenido dentro de los JSON (Elaboración propia).

Una vez creado y terminado de implementar el primer JSON, realizamos cuarenta y nueve copias en las que modificamos únicamente el número del NFT dentro del nombre, y también nos aseguraremos que los nombres de los archivos JSON vayan numerados del uno al cincuenta.

Es importante dicha numeración, ya que una vez lancemos el contrato inteligente, este tendrá que diferenciar a que archivo ha de acceder a partir del nombre del JSON.



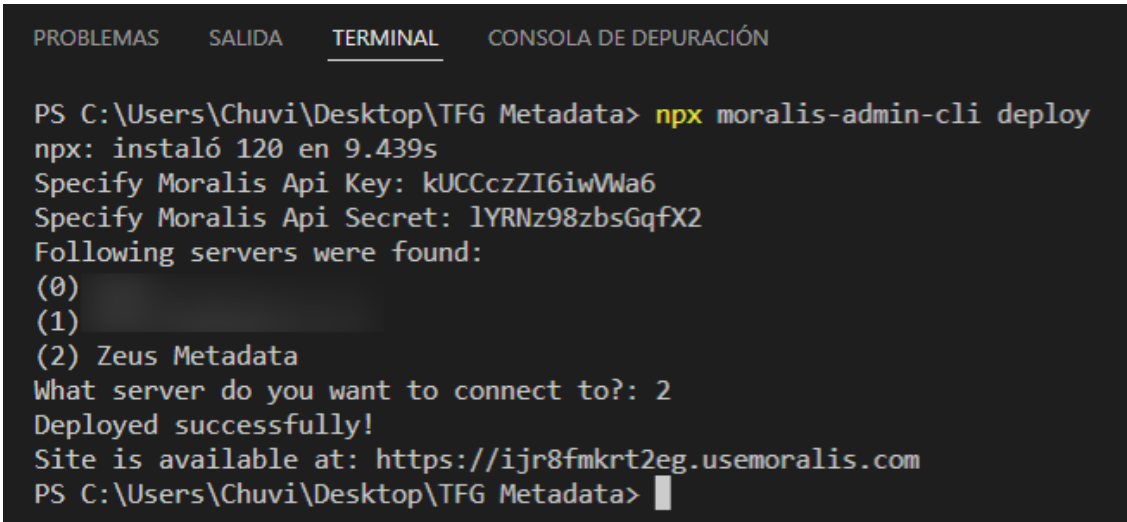
```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda  1.json - TFG Metadata - Visual Studio Code
EXPLORADOR
TFG METADATA
  Imagen
  zeus.png
  Metadata
    1.json
    2.json
    3.json
    4.json
    5.json
    6.json
    7.json
    8.json
    9.json
    10.json
    11.json
    12.json
    13.json
    14.json
    15.json
    16.json
    17.json
    18.json
    19.json
    20.json
ESQUEMA
LÍNEA DE TIEMPO
Terminal
1.json > image
1  {
2    "name": "Zeus #1",
3    "description": "NFT de la colección exclusiva Zeus.",
4    "image": "https://ijr8fmkrt2eg.usemoralis.com/zeus.png"
5  }
```

Figura 5.3: Proyecto que se sube al servidor (Elaboración propia).

Cuando ya tengamos todos los archivos JSON terminados, importamos también la imagen de los NFTs al proyecto. A continuación, procederemos a subir dicho proyecto al servidor que hemos creado con Moralis, una empresa que ofrece entre otros servicios la creación gratuita de servidores.

La subida se ejecuta desde Visual Studio Code, a través de la consola, para ello utilizamos el siguiente comando: `npx moralis-admin-cli deploy`.

Nos solicita el Api Key y el Api Secret de nuestra cuenta, introducimos la información que se nos pide, y seguidamente nos pregunta en cual de los servidores que tenemos asociados a nuestra cuenta es en el que deseamos subir el proyecto.



```
PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN

PS C:\Users\Chuvi\Desktop\TFG Metadata> npx moralis-admin-cli deploy
npx: instaló 120 en 9.439s
Specify Moralis Api Key: kUCCczZI6iwWwA6
Specify Moralis Api Secret: 1YRNz98zbsGqfX2
Following servers were found:
(0)
(1)
(2) Zeus Metadata
What server do you want to connect to?: 2
Deployed successfully!
Site is available at: https://ijr8fmrkt2eg.usemoralis.com
PS C:\Users\Chuvi\Desktop\TFG Metadata> █
```

Figura 5.4: Consola (Elaboración propia).

Finalmente cuando la subida haya terminado se nos notifica en la consola, y ya podemos continuar con el siguiente apartado.

5.2 Creación y lanzamiento del contrato inteligente

Seguiremos con la creación del contrato inteligente, es aquí donde programamos todas las funciones relacionadas con nuestra colección. Desde la función de mintear, es decir, la función para realizar una primera compra de un NFT, hasta la función de poder retirar la cantidad de Ethereums que existan en la dirección en la que lancemos el contrato.

En esta sección haremos hincapié en las partes del código más relevantes, y explicaremos aquellas que tengan mayor dificultad.

Recordamos que estamos utilizando la plataforma online Remix para programar nuestro contrato inteligente. Donde para empezar a escribir código tan solo tenemos que crear un archivo ".sol", y ya podemos programar bajo el lenguaje de Solidity.

Comenzamos declarando las siguientes constantes:

- **cost:** En esta primera variable almacenaremos el coste que va a tener inicialmente los NFTs de la colección, una vez se haga esta primera compra el dueño podrá revender dicho NFT por el valor que desee. En este coste no viene incluido las tasas de la blockchain de Ethereum. Le asignaremos un valor de 0.02 Ethers.

- **maxSupply:** Es el número total de NFTs que se van a vender, esta constante nos servirá para poder finalizar el mintage una vez se agoten. Como la colección cuenta con cincuenta NFTs le asignaremos un valor igual a cincuenta.
- **maxMintAmount:** Como ya hemos explicado limitaremos el número de NFTs que pueda comprar una billetera con el fin de evitar la manipulación de mercado, para ello crearemos esta variable y la asignaremos a cinco, para que como máximo una billetera pueda comprar cinco NFTs de la colección.
- **baseURI:** Almacenaremos en esta constante la dirección donde se almacena la metadata de la colección.

```
uint256 public cost = 0.02 ether;
uint256 public maxSupply = 50;
uint256 public maxMintAmount = 5;
string public baseURI = "https://ijr8fmkrt2eg.usemoralis.com/";
```

Figura 5.5: Variables del contrato inteligente (Elaboración propia).

Una de las funciones más importantes del contrato es la de poder mintagear, como podéis comprobar para poder ejecutar esta función realiza unas comprobaciones previas: Que el número de NFTs que desee mintagear sea inferior o igual a maxMintAmount, también comprueba que la compra no supere el número restante de NFTs que quedan por mintagear, y comprueba que el dueño de la colección no haya pausado el mintageo, ya que podría ejecutar dicha orden por problemas técnicos.

```
function mint (address _to, uint256 _mintAmount) public payable {
    uint256 supply = totalSupply();
    require(!paused);
    require(_mintAmount > 0);
    require(_mintAmount <= maxMintAmount);
    require(supply + _mintAmount <= maxSupply);

    if(msg.sender != owner()) {
        require(msg.value >= cost * _mintAmount);
    }

    for (uint256 i = 1; i <= _mintAmount; i++) {
        _safeMint(_to, supply + i);
    }
}
```

Figura 5.6: Función de mintagear (Elaboración propia).

Existen también algunas funciones que solo puede ejecutar el dueño: Cambiar el coste del mintageo, cambiar el número máximo de mintageo por billetera, cambiar la dirección de donde se extrae la metadata de la colección, poder retirar los fondos...

```
function setCost(uint256 _newCost) public onlyOwner() {
    cost = _newCost;
}

function setmaxMintAmount(uint256 _newmaxMintAmount) public onlyOwner() {
    maxMintAmount = _newmaxMintAmount;
}

function setBaseURI(string memory _newBaseURI) public onlyOwner {
    baseURI = _newBaseURI;
}

function setBaseExtension(string memory _newBaseExtension) public onlyOwner {
    baseExtension = _newBaseExtension;
}

function pause(bool _state) public onlyOwner {
    paused = _state;
}

function withdraw() public payable onlyOwner {
    require(payable(msg.sender).send(address(this).balance));
}
```

Figura 5.7: Funciones privadas (Elaboración propia).

Una vez hayamos finalizado la programación del contrato inteligente, lo compilamos y lo lanzamos en la red de pruebas de Ethereum, Rinkeby. Para el proyecto final lo lanzaremos en la main net, pero para el trabajo final de carrera nos ahorraremos los costes monetarios que esto conlleva.

Para ello tendremos que tener descargada una billetera en nuestro navegador, en mi caso he utilizado MetaMask, estar en la red de Rinkeby, y dentro de Remix lanzar el contrato con la siguiente configuración:

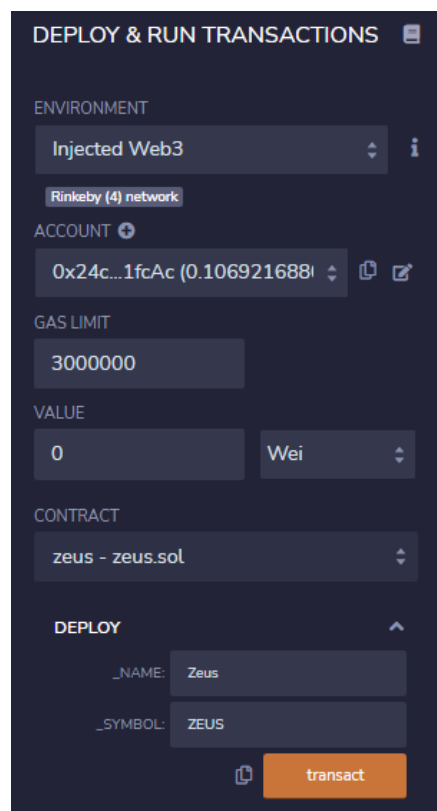


Figura 5.8: Deploy (Elaboración propia).

Cuando ejecutamos la acción de lanzar el contrato dentro de la blockchain, recibimos en mi MetaMask una orden de pago para poder efectuar dicha orden. Cuando el pago se verifique nos aparecerá en la consola de Remix el hash de la transacción.

Esto confirma que nuestro contrato ya se encuentra dentro de la cadena de bloques de Ethereum.

```
[block:10663846 txIndex:11] from: 0x24c...1fcAc to: zeus.(constructor) value: 0 wei data: 0x608...0000 logs: 1 hash: 0xf2d...622e6
status true Transaction mined and execution succeed
transaction hash 0xb23d4e0c17c535c95750c432deb220deaad845eb494c2a7e7271ba09b2cba14a
from 0x24cDbEccAD4bEF0c4a8C13187E7a310d0fd1fcAc
to zeus.(constructor)
gas 3883407 gas
transaction cost 3883407 gas
input 0x608...0000
decoded input {
  "string_name": "Zeus",
  "string_symbol": "ZEUS"
}
decoded output -
logs [
  {
    "from": "0xA486c6dc5E986E9Bdd03081a1c67e2Dcb3AE3942",
    "topic": "0x8bbe079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e0",
    "event": "OwnershipTransferred",
    "args": {
      "0": "0x0000000000000000000000000000000000000000",
      "1": "0x24cDbEccAD4bEF0c4a8C13187E7a310d0fd1fcAc",
      "previousOwner": "0x0000000000000000000000000000000000000000",
      "newOwner": "0x24cDbEccAD4bEF0c4a8C13187E7a310d0fd1fcAc"
    }
  }
]
val 0 wei
```

Figura 5.9: Transacción aceptada (Elaboración propia).

Podemos copiar el hash y visitar la herramienta Etherscan para poder visualizar la información sobre la transacción. Entre otras cosas podemos ver quien es el creador de la transacción, cuál ha sido el coste de esta, la dirección del contrato...

The screenshot shows the Etherscan interface for a transaction on the Rinkeby Testnet. The transaction is successful and has 6 block confirmations. The transaction hash is 0xb23d4e0c17c535c95750c432deb220deaad845eb494c2a7e7271ba09b2cba14a. The transaction was sent from 0x24cDbEccAD4bEF0c4a8C13187E7a310d0fd1fcAc to the contract address 0xa486c6dc5E986E9Bdd03081a1c67e2Dcb3AE3942. The value of the transaction is 0 Ether (\$0.00), and the transaction fee is 0.016903434092586525 Ether (\$0.00). The gas price is 0.000000004352733075 Ether (4.352733075 Gwei).

Figura 5.10: Detalles de la transacción (Elaboración propia).

Si clicamos sobre la dirección del contrato nos redirige a la página del mismo donde podemos visualizar todos los movimientos de este. En este caso como se acaba de crear tan solo aparece dicho movimiento, pero cuando se empiece a vender los primeros NFTs empezarán a guardarse todas las transacciones en esta plataforma.

The screenshot displays the Etherscan interface for a specific contract. At the top, the Etherscan logo and search bar are visible. The contract address is 0xA486c6dc5E986E9Bdd03081a1c67e2Dcb3AE3942. The 'Contract Overview' section shows a balance of 0 Ether. The 'More Info' section indicates that the 'My Name Tag' is 'Not Available' and the 'Creator' is 0x24cdbecad4bef0c4a8... at transaction 0xb23d4e0c17c535c957... The 'Transactions' tab is active, showing a table with one transaction: 'Contract Creation' from 0x24cdbecad4bef0c4a8... at block 10663846, 3 minutes ago, with a value of 0 Ether and a transaction fee of 0.016903434092 Ether.

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0xb23d4e0c17c535c957...	0x60806040	10663846	3 mins ago	0x24cdbecad4bef0c4a8...	IN Contract Creation	0 Ether	0.016903434092

Figura 5.11: Detalles del contrato (Elaboración propia).

Una vez hayamos realizado todos los pasos, ya tendremos nuestra colección dentro de la blockchain de Ethereum. Ahora solo faltará implementar la plataforma desde donde vamos a permitir el mintage con el contrato inteligente, esto lo realizaremos en el capítulo seis.

Pero antes vamos a listar esta colección dentro del mercado de NFTs de Ethereum con más volumen, estamos hablando de OpenSea, para que una vez mintado toda la colección, los dueños puedan vender sus NFTs a través de esta plataforma.

5.3 Listar la colección en el mercado secundario

Nos dirigimos a la plataforma de OpenSea, esta plataforma cuenta con dos mercados, el principal soportado por la main net de Ethereum, y el secundario soportado por cualquiera de las redes de pruebas de Ethereum.

Trabajamos sobre el mercado de OpenSea de Rinkeby, y visitamos la página desde donde nos permite listar nuestra colección. La plataforma nos solicita la dirección del contrato donde se encuentra nuestra colección.

OpenSea Testnets

Search items, collections, and accounts Explore Stats Resources Create

Enter your contract address

What is the address of your ERC721 or ERC1155 contract on the testnet Network?

Rinkeby

Submit

Stay in the loop
Join our mailing list to stay in the loop with our newest feature releases, NFT drops, and tips and tricks for navigating OpenSea.

Your email address Sign up

Join the community

Twitter Instagram Discord Twitch YouTube TikTok Email

OpenSea Marketplace My Account Resources Company
All NFTs Profile Help Center About

Figura 5.12: Listar colección en OpenSea (Elaboración propia).

Una vez se introduce la dirección y la plataforma nos no advierte de ningún error, ya tendremos nuestra colección listada dentro de OpenSea, desde donde cualquiera persona puede realizar ofertas a los tokens ya minteados.

OpenSea Testnets

Search items, collections, and accounts Explore Stats Resources Create

+ Watchlist

Zeus - sMi0cqJDLD

1 item 1 owner floor price 0.00 volume traded

Welcome to the home of Zeus - sMi0cqJDLD on OpenSea. Discover the best items in this collection.

Items Activity

Filter

Status

Buy Now On Auction
New Has Offers
Buy with Card

Price

United States Dollar (USD)

Min to Max Apply

Search by name or trait Single items Price: Low to High

1 item

Zeus - sMi0cqJDLD
Zeus #1

Figura 5.13: Colección listada en OpenSea (Elaboración propia).

Para poder listar una colección necesitamos que como mínimo ya haya habido un minteo, para ello en el constructor del contrato establecimos que el dueño recibiera uno.

Por lo que el NFT que aparece dentro de la colección de OpenSea es el primer NFT de la colección que le pertenece al dueño.

CAPÍTULO 6

Implantación

En este capítulo es donde implementaremos nuestra plataforma, es decir, le añadiremos la posibilidad de mintear nuestra colección. Para ello tendremos que conectarla a nuestro contrato inteligente.

Para la realización de esta utilizamos la herramienta denominada Webflow, la cual está enfocada en un ecosistema bajo la filosofía de no-code, por lo que la inserción de código no está muy bien optimizada.

De hecho se trata de código JavaScript embebido en HTML. A continuación explicaremos las partes del código más relevantes.

Empezaremos con la declaración de las constantes:

```
const CONTRACT_ADDRESS = "0xA486c6dc5E986E9Bdd03081a1c67e2Dcb3AE3942";
const CHAIN_ID = '0x4';
const CHAIN_ID_NAME = 'Rinkeby';
const mintAmount = document.querySelector("#mint-amount");
const messageText = document.querySelector(".message-text");
const numMinted = document.querySelector(".num-minted");
const maxSupply = document.querySelector(".max-supply");
```

Figura 6.1: Constantes de la plataforma (Elaboración propia).

- **CONTRACT ADDRESS:** Pegaremos la dirección del contrato que hemos generado en el anterior capítulo en esta constante.
- **CHAIN ID:** Una de las especificaciones que tenía la plataforma era que reconociera en que red se encontraba la billetera del usuario, y si no era la correcta le pedía que se cambiará. Cada blockchain tiene su ID, en el caso de Rinkeby se trata de la ID "0X4". Esta constante la utilizaremos para comparar si la ID en la que se encuentra el usuario coincide con la de Rinkeby, si no fuera el caso le pediremos que se cambie.
- **CHAIN ID NAME:** Guardaremos en esta variable el nombre de la red en la que se encuentra nuestra colección, en nuestro caso Rinkeby.
- **mintAmount** Dentro del diseño de la página hemos generado un formulario para introducir el número de NFTs que el usuario desea mintear, se almacenará en esta variable.

- **messageText** Se trata de un texto que irá variando dependiendo de las acciones que ejecute el usuario, le advertirá si se encuentra en otra red, le felicitará por la transacción...
- **numMinted** En esta variable iremos almacenando el número de NFTs que se hayan minteado.
- **maxSupply** En esta constante almacenaremos el número total de NFTs que se van a comercializar.

A continuación importaremos una librería de Web3 para poder utilizar funciones que son vitales para nuestro proyecto. Y crearemos una variable donde guardaremos nuestro contrato a partir del ABI y la dirección del contrato:

```
const web3 = new Web3(Web3.givenProvider);  
var contract = new web3.eth.Contract(ABI, CONTRACT_ADDRESS);
```

Figura 6.2: Importar librería Web3 (Elaboración propia).

Cada vez que un usuario cargue la plataforma se le tendrá que actualizar la variable en la que se guardan cuantos NFTs se han minteando y cual es el máximo que se van a mintear. Por lo que utilizaremos los siguientes métodos:

```
contract.methods.totalSupply().call().then((supply) => numMinted.innerText = supply);  
contract.methods.maxSupply().call().then((max) => maxSupply.innerText = max);
```

Figura 6.3: Actualización de variables (Elaboración propia).

La primera función importante que encontramos dentro del código es la de la comprobación de la red en la que se encuentra el usuario, el cuál será notificado si se encuentra en la red correcta o no:

```
const updateChainId = async function () {  
  const id = await ethereum.request({ method: 'eth_chainId' });  
  console.log('id from async func is: ' + id);  
  
  if (id == CHAIN_ID) {  
    console.log('chain is' + CHAIN_ID_NAME);  
    messageText.innerText = "👍 You are connected to " + CHAIN_ID_NAME;  
    mintForm.classList.remove("disabled");  
  } else {  
    console.log('chain is not ' + CHAIN_ID_NAME);  
    messageText.innerText = "Please switch to " + CHAIN_ID_NAME;  
    mintForm.classList.add("disabled");  
  }  
  return id;  
}
```

Figura 6.4: Función que comprueba la red en la que te encuentras (Elaboración propia).

Si continuamos encontramos la función más importante del código, que es la función de mintear:

```
const mint = async function (amount) {
  // get a list of accounts from metamask
  const accounts = await window.ethereum.request({
    method: "eth_requestAccounts",
  });
  var totalCost = amount;
  console.log(totalCost);
  // if an account is present
  if (accounts.length > 0) {
    contract.methods.mint(accounts[0], amount).send({
      from: accounts[0],
      to: CONTRACT_ADDRESS,
      value: String(totalCost),
    });
  }
}
```

Figura 6.5: Función de mintear en la plataforma (Elaboración propia).

También tuvimos que insertar las siguientes líneas de código para que pudiéramos utilizar un servicio denominado Crossmint, el cual con una configuración previa desde su plataforma permite realizar la compra de nuestra colección a través de una transferencia bancaria.

Esta configuración que acabo de mencionar se trataba únicamente de rellenar un formulario cuyos parámetros eran la dirección del contrato y el coste del minteo.

```
<script src="https://unpkg.com/@crossmint/client-sdk-vanilla-ui@0.0.1-alpha.1/lib/index.global.js"></script>
<crossmint-pay-button
  clientId="797a8332-b984-403d-b73e-2f18e4049576"
  environment="staging"
  mintConfig='{ "type": "erc-721", "price": "0.02", "_mintAmount": "1" }'
/>
```

Figura 6.6: Código relacionado con Crossmint (Elaboración propia).

Por último, implementamos a partir del API de OpenSea una función que permitía recuperar todos los tokens que poseía una billetera. Esto nos servía para que la plataforma cumpliera con el objetivo de mostrar los NFTs de la billetera que esté conectada a nuestra plataforma.

Una vez finalizado toda la implementación de la plataforma, es hora de comprobar que todo lo realizado en este trabajo final de carrera funciona de manera correcta a través de la realización de pruebas.

CAPÍTULO 7

Pruebas

Como mencioné en el capítulo uno, este trabajo ha sido realizado de manera conjunta con un compañero, donde él se encarga de realizar el diseño de la plataforma, y este ha sido el resultado:

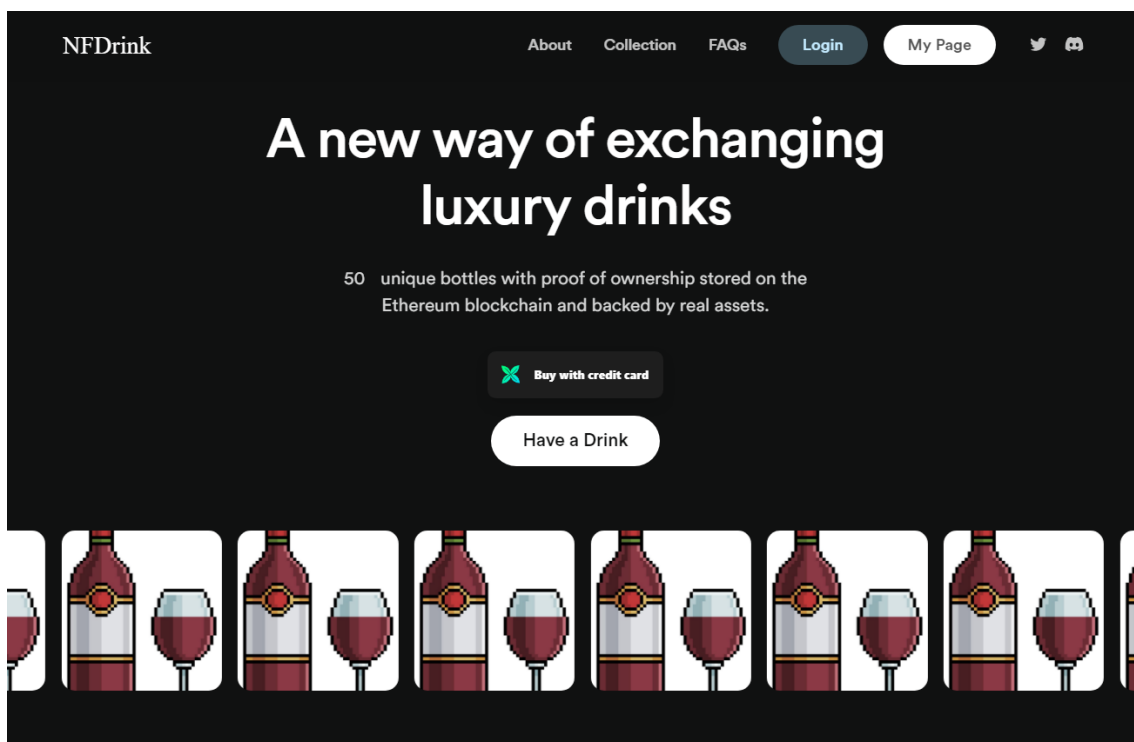


Figura 7.1: Diseño de la plataforma (Elaboración propia).

En este capítulo vamos a verificar que la plataforma que hemos creado junto a todos los elementos que hemos configurado, funcionan de una manera correcta y armónica con el diseño creado por mi compañero. Para ello comprobaremos las siguientes funcionalidades y cualidades de la plataforma:

- El usuario puede iniciar sesión.
- El usuario puede comprar un NFT a través de su billetera.
- El usuario puede comprar un NFT a través de la cuenta del banco.
- El usuario puede visualizar sus NFTs.

7.1 Iniciar sesión

Vamos a simular que somos un cliente y queremos registrarnos en la plataforma, para ello clicamos sobre el botón "Login".

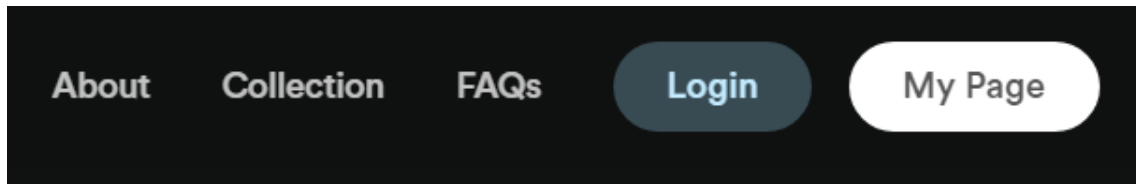


Figura 7.2: Barra de navegación - Login (Elaboración propia).

Se nos abrirá una ventana emergente, es decir, un pop up. Donde nos dará tres opciones para realizar el inicio de sesión: a través de Google, a través de un correo y una contraseña, o a través de una billetera.

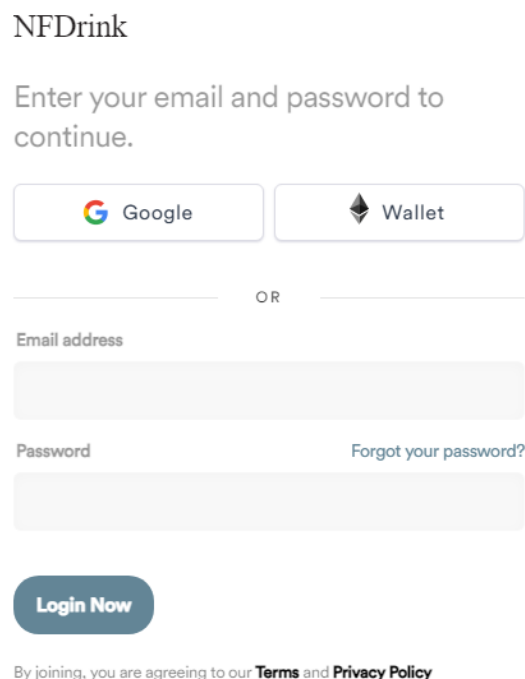
The login form for NFDrink. It starts with the brand name 'NFDrink'. Below it, the instruction 'Enter your email and password to continue.' is shown. There are two buttons for social login: 'Google' and 'Wallet'. A horizontal line with 'OR' in the center separates these from the standard login fields. The 'Email address' field is a wide text input. Below it is the 'Password' field, which is also wide. To the right of the password field is a link that says 'Forgot your password?'. At the bottom of the form is a dark blue button with the text 'Login Now'. Below the button, there is a small line of text: 'By joining, you are agreeing to our Terms and Privacy Policy'.

Figura 7.3: Inicio de sesión (Elaboración propia).

Como la configuración de inicio de sesión a través de la billetera fue la más costosa, elegiremos este método para continuar con las pruebas.

En esta ocasión utilizamos la billetera MetaMask, nos vamos a la red principal de Ethereum, para que la plataforma nos diga que nos tenemos que cambiar a la de Rinkeby, ya que es aquí donde se encuentra la colección.

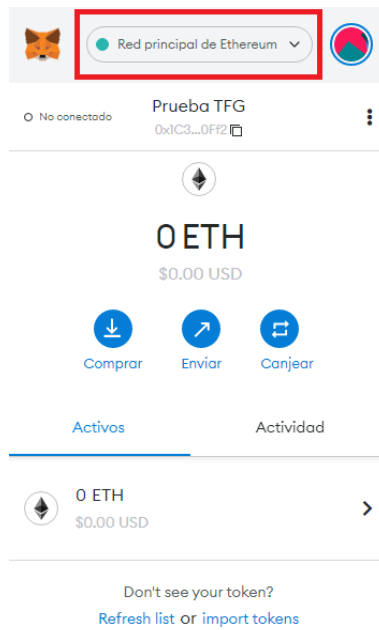


Figura 7.4: MetaMask (Elaboración propia).

Y ahora si clicamos sobre el botón "Wallet", se nos abrirá un nuevo pop up, esta vez de la aplicación de MetaMask, donde nos advierte de que nos encontramos en una red equivocada, y nos solicita cambiar a la correcta.



Figura 7.5: Advertencia cambiar de red (Elaboración propia).

Al cambiar de red podemos iniciar sesión, y ya podemos utilizar el resto de funcionalidades de la plataforma.

7.2 Comprar NFT a través de una billetera

Vamos a proceder a realizar una prueba de compra a través de una billetera. En mi caso voy a seguir utilizando MetaMask para realizar esta transacción.

Comenzamos yendo a la página principal de la plataforma, y clicamos sobre "Have a Drink", nos pide el número de NFTs que deseamos mintear, como podemos comprobar el máximo es cinco NFTs, tal como establecimos anteriormente, para este caso vamos a seleccionar únicamente uno.

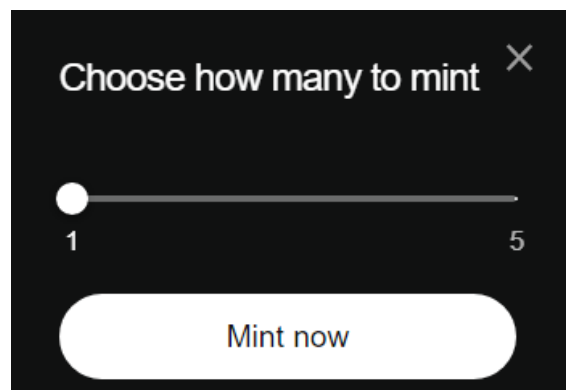


Figura 7.6: Botón para mintear (Elaboración propia).

Al clicar sobre "Mint now" se nos abre una ventana de MetaMask donde podemos encontrar la información sobre la transacción que está a punto de ejecutarse, es decir, podemos ver el coste del minteo, el coste del gas, y más información relevante sobre la transacción.

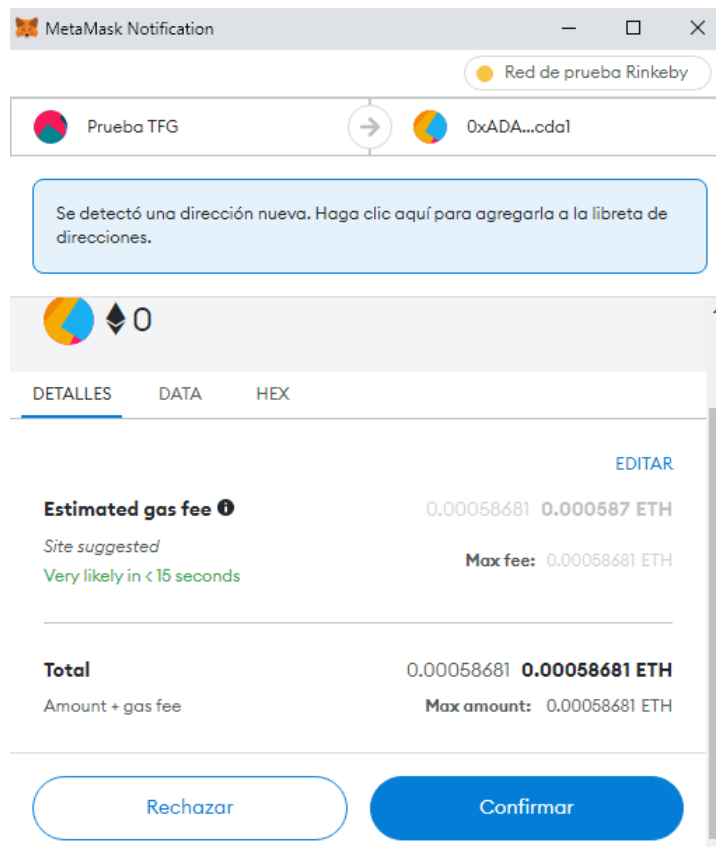
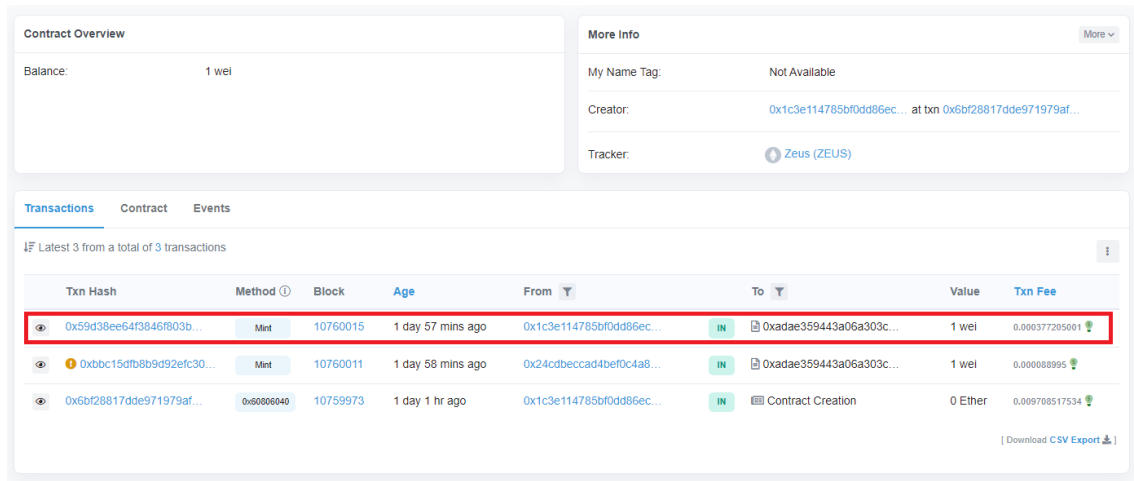


Figura 7.7: Transacción MetaMask (Elaboración propia).

Una vez aceptemos dicha transacción, la blockchain la tratará, y en cuestión de segundos será aceptada. Cuando haya terminado de ejecutarse, en nuestro historial de transacciones podremos visualizar que efectivamente el minto ha sido aceptado.

Para verificar que esto es correcto, nos vamos a EtherScan y en la dirección del contrato de la colección comprobamos que efectivamente ha habido un minto, cuyo dueño tiene que concordar con nuestra billetera.



Contract Overview

Balance: 1 wei

More Info

My Name Tag: Not Available

Creator: 0x1c3e114785bf0dd86ec... at txn 0x6bf28817dde971979af...

Tracker: Zeus (ZEUS)

Transactions

Latest 3 from a total of 3 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x59d38ee64f3846f803b...	Mint	10760015	1 day 57 mins ago	0x1c3e114785bf0dd86ec...	IN 0xad359443a06a303c...	1 wei	0.000377205001
0xbbc15dfb8b9d92efc30...	Mint	10760011	1 day 58 mins ago	0x24cdbeccad4bef0c4a8...	IN 0xad359443a06a303c...	1 wei	0.000088995
0x6bf28817dde971979af...	0x60806040	10759973	1 day 1 hr ago	0x1c3e114785bf0dd86ec...	IN Contract Creation	0 Ether	0.009708517534

[Download CSV Export]

Figura 7.8: Transacciones del contrato (Elaboración propia).

Para verificar todavía más esta transacción, vamos a OpenSea y en nuestro perfil vamos a comprobar que realmente se encuentra el NFT que acabamos de comprar y podemos hacer lo que queramos con el.

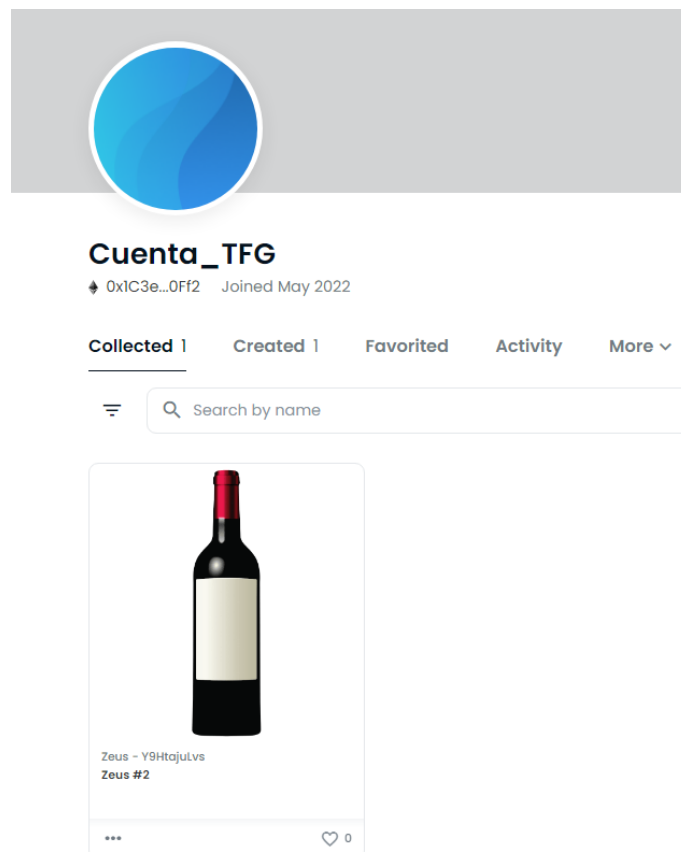


Figura 7.9: Perfil en OpenSea (Elaboración propia).

7.3 Comprar NFT a través de la cuenta del banco

Para conseguir esta funcionalidad utilizamos el servicio de Crossmint, se trata de un plataforma que calcula el precio del NFT en dólares, teniendo en cuenta su valor original más los gastos de la transacción, y permite al usuario realizar la compra del NFT a través de una pasarela de pago.

Realmente Crossmint coge tu dinero y lo transforma en ETH, y compra por ti el NFT. Una vez el usuario haya realizado el pago, Crossmint genera una billetera automática para el cliente y es allí donde se encontrará el NFT que acaba de comprar.

Su implementación fue muy sencilla, ya que tan solo tuvimos que especificar la dirección del contrato y el coste al que se lanzaba la colección.

A continuación vamos a realizar la compra de un NFT a través de la pasarela de pago de Crossmint, para ello nos dirigimos a la página principal de la plataforma y clicamos sobre el botón de Crossmint.

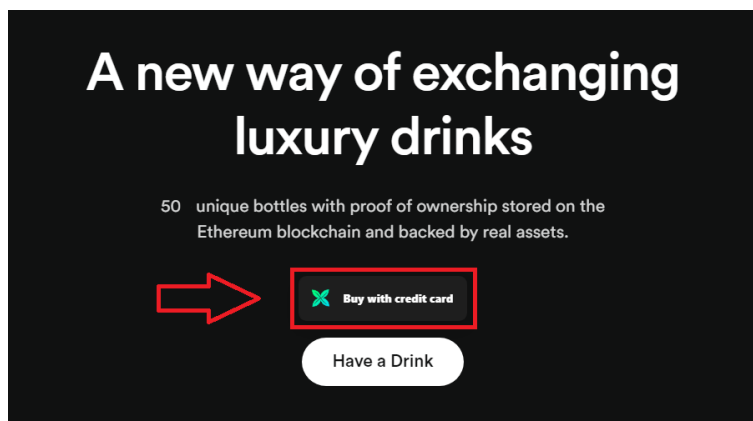


Figura 7.10: Botón para comprar a través del banco (Elaboración propia).

Nos redirigirá a una pasarela de pago, y es aquí donde tendremos que rellenar el formulario con nuestros datos bancarios.

Figura 7.11: Pasarela de pago (Elaboración propia).

Una de las grandes cualidades de este servicio es que te permite realizar pruebas en la red de Rinkeby, ya que te facilitan una tarjeta bancaria para que el coste de la transacción sea cero.

Una vez el pago se haya efectuado, recibiremos un email de confirmación con un enlace que nos llevará a la billetera que se nos ha creado y donde podemos ver el NFT que acabamos de comprar.



Figura 7.12: Cartera en Crossmint (Elaboración propia).

Podríamos verificar que esta transacción se ha llevado a cabo en la dirección del contrato en EtherScan, y comprobar que realmente ha habido un movimiento de compra.

7.4 Visualizar los NFTs

Uno de los objetivos que nos propusimos para que nuestra colección diera un paso más que el resto, es que el usuario pudiera ver sus NFTs a través de nuestra plataforma.

Vamos a comprobar si realmente esta funcionalidad está bien implementada, para ello nos dirigimos a la página llamada "My Page".

Si el usuario ya tiene conectado su billetera con la plataforma y la funcionalidad está bien desarrollada, se debería de mostrar todos los NFTs que posee la billetera que está conectada.

Como previamente habíamos realizado una compra con nuestra billetera a la hora de realizar la prueba, nos va a aparecer el NFT que hemos comprado.

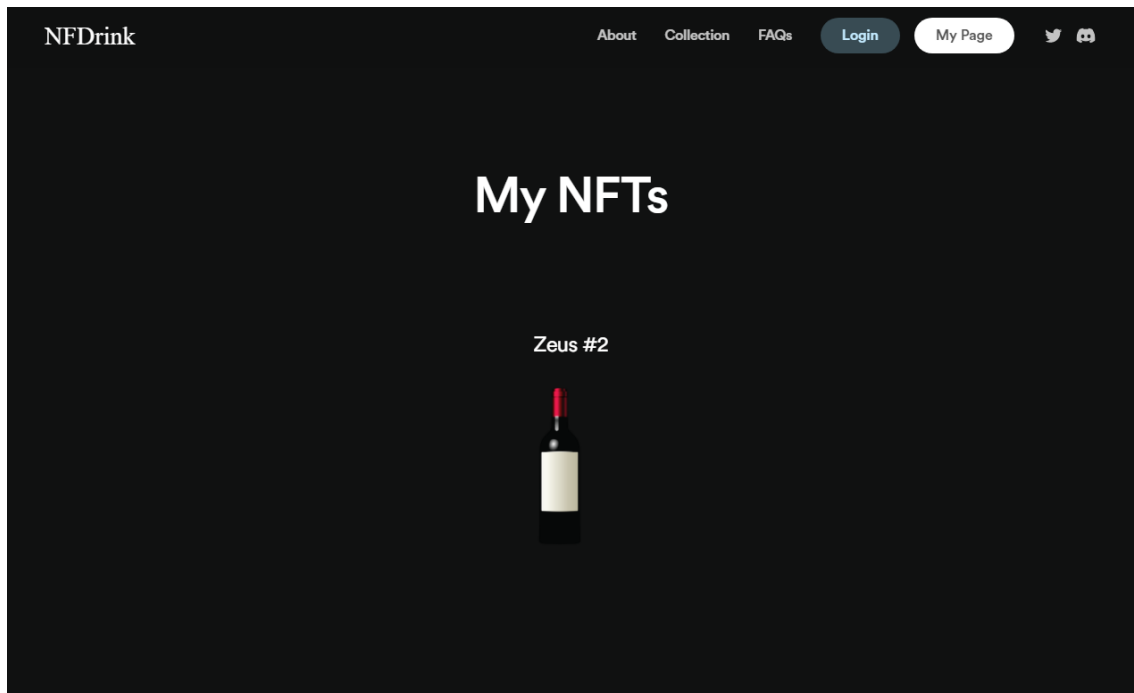


Figura 7.13: Ver NFTs de una billetera (Elaboración propia).

CAPÍTULO 8

Conclusiones

Una vez finalizado el proyecto es hora de ver si hemos cumplido con los objetivos propuestos en la introducción, para ello mencionaremos uno a uno y comentaremos si lo hemos podido ejecutar y que problemas e inconvenientes hemos tenido en su realización.

- **Llevar a cabo una interfaz que sea fácil e intuitiva para el usuario principiante:** Como hemos podido ver en el anterior capítulo hemos conseguido que la plataforma tenga un diseño muy profesional y, a la vez sea fácil de utilizar para aquellas personas que no saben tanto sobre este mundo.
- **El usuario podrá registrarse a través de su billetera o correo electrónico:** Hemos implementado un sistema de registro e inicio de sesión de manera exitosa, además para aquellas personas que se registren, tienen la opción de poder disfrutar de todas las funcionalidades de la plataforma.
- **Si a la hora de realizar la compra, la billetera no se encuentra en la red de Ethereum la plataforma se lo notificará:** Si un usuario conecta su billetera y esta se encuentra en una red distinta a la de la colección, la plataforma le notifica de esto, y además le dice a cuál red se tiene que cambiar para poder comprar un NFT de la colección.
- **Permitir realizar la compra a través de una transacción bancaria:** Gracias a Crossmint hemos conseguido mediante la inserción de dos líneas de código, la posibilidad de que aquellos usuarios menos experimentados puedan comprar a través de su banco nuestra colección.
- **Una vez el usuario haya adquirido su NFT siempre tendrá la opción de quemarlo:** Para conseguir este objetivo creamos un apartado para que aquellos dueños que desearan quemar su NFT lo pudieran hacer.
- **Existirá una página donde permitirá visualizar al usuario los NFTs que le pertenecen.** Aquellas personas que se registren tienen la opción de poder visualizar sus NFTs a través de nuestra plataforma.

8.1 Relación del trabajo desarrollado con los estudios cursados

Es cierto que en la carrera hemos podido aprender muchos lenguajes de programación que han ayudado a que pueda ejecutar este trabajo, pero también ha sido muy importante la formación que he recibido de forma autodidacta a través de distintas plataformas de enseñanza.

Gracias a lo aprendido en la carrera tengo la facilidad de poder aprender lenguajes de programación de manera mucho más rápida y eficiente, como en este caso he podido aprender sobre Solidity y el lanzamiento de contratos inteligentes.

También este trabajo me ha ayudado a comprender y entender mejor lo que es web3, ya que una gran parte del código que he realizado iba sobre esto.

8.2 Trabajos futuros

Como he podido explicar en el capítulo cinco, este trabajo se trata de una prueba para un proyecto que queremos lanzar de cara al próximo invierno. Donde realmente la parte tecnológica va a ser la misma, aunque auditaremos el código e intentaremos mejorarlo.

Por lo que nuestro siguiente paso es empezar con el tema de las redes sociales y el marketing, ya que el producto que deseamos vender ya lo tenemos y consideramos que es muy bueno, pero ahora falta lo más difícil de este mundo, que es persuadir el cliente a través del marketing.

Intentaremos que nuestras redes sociales vayan aumentando en número gracias a un buen contenido, e intentaremos conseguir acuerdos comerciales con influencers del sector de la blockchain.

Bibliografía

- [1] Cadena de bloques. (24 de junio de 2022). En Wikipedia. https://es.wikipedia.org/wiki/Cadena_de_bloques.
- [2] Imagen extraída el 24 de mayo de 2022. Consultada en <https://www.iebschool.com/blog/blockchain-cadena-bloques-revoluciona-sector-financiero-finanzas/>.
- [3] Imagen extraída el 26 de mayo de 2022. Consultada en <https://opensea.io/assets/ethereum/0xbc4ca0eda7647a8ab7c2061c2e118a18a936f13d/9719>.
- [4] Olivares, Pedro. (2022). Comparación entre bebidas de lujo y el S&P500 [Digital].
- [5] McEvoy, Jake. (08 de febrero de 2022). ¿Cuándo comenzaron las NFT?. KYM. <https://www.knowyourmobile.com/es/noticias/cuando-empez%C3%B3-nfts/>.
- [6] Cruz, Danielle. (10 de junio de 2021). El primer NFT de la historia vendido. Elemental. <https://elemental.com/2021/06/10/el-primer-nft-de-la-historia-vendido-por-casi-15-millon-de-dolares-en-sothebys/>.
- [7] Imagen extraída el 27 de mayo de 2022. Consultada en <https://www.coindesk.com/news/articleView.html?idxno=77217>
- [8] Imagen extraída el 2 de junio de 2022. Consultada en <https://cryptopunks.app/cryptopunks/details/5822>
- [9] Imagen extraída el 3 de junio de 2022. Consultada en <https://blockbar.com/>

APÉNDICE A

Objetivos de desarrollo sostenible

Objetivos de desarrollo sostenible	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.		X		
ODS 4. Educación de calidad.		X		
ODS 5. Igualdad de género.		X		
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.	X			
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.		X		
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.		X		
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.		X		
ODS 16. Paz, justicia e instituciones sólidas.		X		
ODS 17. Alianzas para lograr objetivos.		X		

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados

En mi TFG explico que es la blockchain y como funciona, gracias a esta nueva y emergente tecnología se podría conseguir representaciones monetarias a nivel internacional, cuyo objetivo beneficiaría a un menor gasto de energía por parte de todas las organizaciones centralizadas monetarias. Este menor gasto se traduce en una acción global hacia la sostenibilidad del clima.

Esta tecnología también ayudaría a conseguir un mundo más igualitario en todos los aspectos, y un mundo donde todos tendríamos las mismas oportunidades.